

Desenvolvimento de um Programa de Computador para o Dimensionamento de Fornos de Cal Verticais

Anne-Sophie Vilela

Dissertação de Mestrado

Orientador na FEUP: Prof. Carlos Pinho



Mestrado Integrado em Engenharia Mecânica

fevereiro 2016

Resumo

A presente dissertação tem como principal objetivo a criação de um programa de dimensionamento de fornos de cal vertical, que seja bastante versátil e que permita avaliar o comportamento térmico deste tipo de forno sob diferentes condições operacionais, para servir como uma ferramenta auxiliar no projeto e na avaliação da viabilidade técnica. O foco da utilização do programa de simulação desenvolvido neste trabalho é o projeto de um forno vertical para a produção de cal, deste modo, é apresentada uma revisão da utilização da cal.

Assim, uma revisão da utilização da cal é apresentada. O modelo matemático e o funcionamento geral do programa de simulação computacional de fornos verticais desenvolvido neste trabalho são descritos e os resultados das simulações realizadas são apresentados como estudos da geometria e do comportamento térmico do forno vertical.

Finalmente, com base nos dados levantados no decorrer deste trabalho e nos resultados das simulações computacionais são sugeridas algumas recomendações de projeto para o forno vertical para a produção de cal, tendo em vista o aumento da qualidade do processo de calcinação.

Abstract

This work aims at the development of a heat transfer simulation software for shaft kilns, capable of simulating kilns under different operational conditions, to serve as an auxiliary tool in the design and technical evaluation of lime kilns. The use of the simulation software developed in this work is focused in the design of shaft kilns.

Thus, the current usage of lime is reviewed. The mathematical model and the main routines of the shaft kiln simulation program are described, and the results of several simulations are presented as case studies concerning the thermal behavior of the shaft kiln in the proposed production site.

Finally, based on the data collected during this work and in the computational simulation results, some design recommendations for the lime producing shaft kiln are suggested, aiming at enhancing the quality of the lime obtained and reducing the energy consumption.

Agradecimentos

Em primeiro lugar, quero agradecer ao meu orientador, Professor Doutor Carlos Pinho, por todo o seu apoio, disponibilidade e dedicação demonstrada ao longo não só desta dissertação mas também ao longo do meu percurso académico. Ainda um obrigado pela disponibilização de muitos recursos bibliográficos e incentivos permanentes.

Em seguida agradeço ao meu amigo Eng. Bruno Fernandes pela ajuda prestada na construção do programa e esclarecimento de dúvidas de programação. A sua colaboração foi imprescindível na realização desta dissertação.

Agradeço ao meu amigo e colega Rui Ramos pelo companheirismo e pelas longas horas durante esta jornada, no meu percurso académico.

Agradeço aos meus pais, pelo amor incondicional, incentivo e apoio constante a cada dia. Ainda pelo suporte e sustento da minha vida.

Os meus sinceros agradecimentos.

Índice

Resumo	i
Abstract	iii
Agradecimentos	v
Nomenclatura	xvii
Capítulo 1.....	1
Introdução	1
1.1. – Motivação.....	2
1.2. – Objetivo	2
1.3. – Estrutura da dissertação	3
Capítulo 2.....	5
A cal	5
2.1. – Introdução.....	5
2.2. – História	6
2.3. – Origem mineral	7
2.4. – Componentes da cal	8
2.4.1. – Características físico-químicas.....	8
2.5. – Aplicações da cal	9
2.6. – Técnicas de produção	13
2.6.1 – Extração, desmonte e britagem.....	13
2.6.2. – Fornos de calcinação.....	15
2.6.2.3. – Fornos verticais.....	17
Capítulo 3.....	35
Forno vertical	35
3.1. – Introdução.....	35
3.2. – Seleção de forno	38

3.3. – Forno vertical normal.....	39
3.4. – Características geométricas	40
3.5. – Descrição do processo	41
3.6. – Problemas ocorrentes	42
3.6.1. – Baixa eficiência do forno.....	43
3.6.2. – Calcinação incompleta.....	44
3.6.3. – A influência do tipo e tamanho do combustível no desempenho do forno.....	45
3.6.4 – Alto teor de CO nos gases de escape	45
3.7. – Combustível.....	46
3.7.1. – Combustão	48
3.8. – O forno vertical comparativamente ao forno rotativo	49
Capítulo 4.....	51
Calcinação	51
4.1. –Introdução	51
4.2. – Degradação térmica	52
4.3. – Distribuição do tamanho da partícula	53
4.4. – Formato da partícula.....	53
4.5. – Reação química	54
4.6. – Cinética de calcinação.....	55
4.6.1. – Dissociação de calcário com alto teor de cálcio	56
4.7. – Fatores que influenciam a calcinação.....	58
4.7.1. – Propriedades da carga do forno	59
4.7.2. – Desempenho do forno/reação físico-química	59
4.7.3. – Combustível, aproveitamento e distribuição no forno.....	67
4.8. – Sinterização de cal com alto teor de cálcio	68
Capítulo 5.....	71
Modelo de cálculo.....	71
5.1. – Dimensão do forno.....	71
5.1.1. – Dimensão da zona de calcinação	75
5.1.2. – Dimensão da zona de arrefecimento.....	76
5.1.3. – Combustão.....	78
5.1.3.1. – Produtos	80
5.1.3.2. – Propriedades da combustão.....	83
5.1.4. – Dimensão da zona de pré-aquecimento.....	83
5.1.5. – Altura do forno	84
5.3 – Transferência de calor na calcinação	85

5.4 – Perda de carga.....	88
5.5 – Calcinação	89
Capítulo 6.....	91
Programa de simulação computacional	91
6.2. – Programa base	92
6.3. – Apresentação do programa.....	93
6.4 – Funcionamento da aplicação	96
6.4.1 – Entrada de dados	96
6.4.2 – Desenvolvimento	97
6.4.3 – Saída de dados.....	98
Capítulo 7.....	101
Resultados da simulação	101
7.1 – Simulação	101
Capítulo 8.....	107
Conclusões e sugestões.....	107
8.1. – Conclusão	107
8.2. – Trabalhos Futuros	108
Referências.....	111
Anexo A	115
Algoritmo do Programa	115
Anexo B	237
Propriedades físicas	237
Anexo C	239

Lista de figuras

Figura 2.1 - Sequência de fabrico da cal.	13
Figura 2.2 - Operações de extração e desmonte (LNEG, 2015)	14
Figura 2.3 - Forno de barranco em cuba ovoide de alvenaria (Guimarães, 2002).....	16
Figura 2.4 - Tipo de fornos contínuos verticais para a produção de cal (Guimarães, 2002).....	18
Figura 2.5 - Forno de calcinação <i>AZBE</i> (Perona, 1976).....	19
Figura 2.6 - Forno de calcinação <i>AZBE</i> a gás natural, com controlo integral de temperatura na zona de calcinação (Perona, 1976).....	20
Figura 2.7 - Forno de calcinação <i>West</i> . CO ₂ <2%; consumo de 4605,5 kJ/kg; consumo energético de 15 kWh/t de cal; flexibilidade de capacidade de 60% (Perona, 1976).....	21
Figura 2.8 - Forno industrial de calcinação <i>Maerz Ofendau Parallel Shaft</i> (Guimarães, 2002).....	23
Figura 2.9 - Esquema de funcionamento de um forno de calcinação, modelo <i>Parallel shaft</i> , produzido pela <i>Maerz Ofendau</i> (Guimarães, 2002).....	24
Figura 2.10 - Forno para a calcinação de finos de calcário, produzido pela <i>F. L Smidth</i> (Smidth, 1975).	25
Figura 2.11 - Forno de calcinação em leito fluidizado da <i>Kawasaki</i> (Kawasaki, 1983).	26
Figura 2.12 - Esquema de forno horizontal de câmara rotativa desenvolvido por <i>Calcinatic International Limited</i> (Kinsler, 1991).	27
Figura 2.13 - Planta esquemática do forno <i>Hoffman</i> (Boynton, 1980).....	28
Figura 2.14 - Forno calcinador <i>DSO</i> (Schwarzkorf, 1978).....	29

Figura 2.15 - Forno calcinador <i>Hereshoff</i> (Schwarzkorf, 1978).	30
Figura 2.16 - Forno calcinador <i>Kunnep Chisaki</i> (Schwarzkorf, 1978).	30
Figura 2.17 - Modelo de forno rotativo da <i>F. L. Smidth</i> , produzido na Dinamarca (Smidth, 1975).	31
Figura 2.18 - Forno rotativo modelo <i>Ransome</i> , de 1885 (Ciaccia <i>et al</i> , 1974). ..	31
Figura 2.19 - Esquema de forno rotativo longo (Manfred <i>et al</i> . 1976).	33
Figura 3.1 - Secção transversal de um forno vertical normal (Oates, 1998). ...	40
Figura 3.2 - Perfil de temperatura num forno vertical.	42
Figura 3.3 - Fotografia de um fragmento de uma escória cristalina com 70 mm de diâmetro (Leslie <i>et al.</i> , 2004).	44
Figura 3.4 - Secção transversal de partículas de cal não calcinadas (Tran <i>et al.</i> , 2005).	45
Figura 4.1 - Variação da pressão de dissociação com a temperatura (Oates, 1998).	54
Figura 4.2 - Ilustração do processo de dissociação de calcário (Oates, 1998). ..	56
Figura 4.3 - Processo de calcinação de uma pedra calcária esférica com 31mm (Oates, 1998).	58
Figura 4.4 - Pedra de cal com núcleo de carbonato não dissociado (Quallical, 2015).	60
Figura 4.5 - Variação do tempo de calcinação com a temperatura e tamanho da pedra calcária. (Azbe, 1953).	61
Figura 4.6 - Gráfico de distribuição de zonas no forno vertical (Guimarães, 2002).	62
Figura 4.7 - Croqui da transformação da estrutura das partículas de cal (Guimarães, 2002).	64
Figura 4.8 - Microscopia eletrónica de varredura de partículas de cal a diferentes condições de calcinação (Guimarães, 2002).	65
Figura 4.9 - Influência da temperatura de calcinação (Millet, 1979).	66
Figura 4.10 - Variação da massa volúmica aparente com a temperatura e tempo (Oates, 1998).	68
Figura 5.1 - Possíveis arranjos de esferas - dois ortorrômbica (Yang, 2003)... ..	73
Figura 5.2 - Esquema da análise do forno.	75

Figura 5.3 - Esquema da análise de um seção do forno.	77
Figura 6.1 - Síntese da elaboração do programa.	92
Figura 6.2 - Interface gráfica da entrada de valores.	94
Figura 6.3 - Interface gráfica de saída de valores.	95
Figura 6.4 - Interface de entrada de dados.	96
Figura 6.5 - Diagrama de funcionamento da aplicação.	98
Figura 6.6 - Interface de resultados.	99
Figura 6.7 - Ficheiro texto dos resultados.	100

Lista de tabelas

Tabela 2.1 - Classificação das rochas carbonatadas cálcio-magnesianas (Guimarães, 2002).....	8
Tabela 2.2 - Características físicas e químicas da cal (Guimarães, 2002).....	9
Tabela 2.3 - Aplicações da cal (Boynton,1980).	11
Tabela 2.4 - Caraterísticas de produção de um forno de calcinação West (Perona, 1976).....	21
Tabela 3.1 - Sumário das características dos tipos de fornos mais comuns (G – Gasoso, L – Líquido, S- Sólido) (European Commission, 2013).	36
Tabela 3.2 - Número de Instalações de produção de cal na União Europeia em 2003 (European Commission, 2013).	37
Tabela 3.3 - Distribuição de combustível usada na queima de cal na EU-27 em 2003 (European Commission, 2013).	38
Tabela 4.1 - Pressão de equilíbrio do CO ₂ na calcinação (Beach, 1983).....	52
Tabela 4.2 - Quantidade de calcário calcinada a cada hora (Predescu, 1989).....	69
Tabela 4.3 - Valores estimados da dissociação e sinterização após 10 h a 1200 °C (Oates, 1998).	70
Tabela 5.1 - Caraterísticas de empacotamento (Yang, 2003).	73
Tabela 5.2 - Composição típica de um gás natural (Trangás).....	79
Tabela 5.3 – Composição dos produtos da combustão.....	80
Tabela 7.1 - Dados de entrada.	102
Tabela 7.2 – Resultados obtidos da simulação.	102
Tabela 7.3 - Comparação dos resultados obtidos.	103

Tabela 7.4 - Comparação de resultados em relação ao excesso de ar na combustão.	104
--	-----

Tabela 7.5 - Valores das propriedades físicas dos gases para diferentes valores de excesso de ar.	105
--	-----

Nomenclatura

A_t	Área seção reta do forno	[m ²]
A_b	Área da base	[m ²]
C_p	Calor específico	[kJ/kg]
C_{p_p}	Calor específico dos produtos de combustão	[kJ/kg]
d_p	Diâmetro da pedra	[m]
D	Diâmetro do forno	[m]
e	Excesso de ar na combustão	
F_{1-1}	Fator de forma	
g	Aceleração gravítica	[m ² /s]
Gr	Número de Grashof	[-]
H	Altura do forno	[m]
H_i	Altura da zona i	[m]
h_{amb}	Coeficiente de transferência de calor no exterior do forno	[W/(m ² K)]
h_{ef}	Coeficiente de transferência de calor efetivo	[W/(m ² K)]
h_i	Coeficiente de transferência de calor da zona i	[W/(m ² K)]
h_{int}	Coeficiente de transferência de calor no interior do forno	[W/(m ² K)]
k_i	Condutividade térmica do fluido (ar) na zona i	[W/(m K)]
Kd	Fator pré-exponencial da taxa de decomposição do calcário	[mol/(m ² atm s)]
k_{g_i}	Condutibilidade térmica do ar ambiente	[W/(m K)]
k	Condutibilidade térmica do gás de combustão	[W/(m K)]
k_{r_1}	Condutibilidade térmica do tijolo refratário 1	[W/(m K)]
k_{r_2}	Condutibilidade térmica do tijolo refratário 2	[W/(m K)]
L_c	Comprimento característico da geometria (altura do forno)	[m]
L_{r_1}	Comprimento do refratário1	[m]
L_{r_2}	Comprimento do refratário2	[m]
\dot{m}_{g_i}	Caudal de fluido (ar) na zona i	[kg/s]
\dot{m}_{comb}	Caudal de combustível	[kg/s]
M_i	Massa molar	[kg/kmol]
\dot{m}_p	Caudal dos produtos da combustão	[kg/s]
m_p	Massa dos produtos da combustão	[kg]
\dot{m}_{CO_2}	Caudal de CO ₂	[kg/s]
\dot{m}_t	Caudal total do fluido na zona de pré-aquecimento	[kg/s]

N_p	Número de esferas por unidade de volume do leito	[-]
n_{pt}	Número de pedras no forno	[-]
n_{pc}	Número de pedras na zona de calcinação	[-]
\overline{Nu}	Número de Nusselt	[-]
PCI	Poder calorífico inferior do combustível	[kJ/kg]
Pr_i	Número de <i>Prandtl</i> na zona i	[-]
P_i	Pressão no interior do forno	[atm]
P_{eq}	Pressão de equilíbrio do CO ₂	[atm]
Q	Quantidade de energia	[W]
\dot{q}	Fluxo de calor na zona de calcinação	[W/m ² K]
R	Raio interior do forno.	[m]
Ra	Número de <i>Rayleigh</i>	[-]
R_c	Taxa de reação de calcinação	[mol/m ² s]
r_{cc}	Rácio cal – calcário	[kg]
r_{cco_2}	Rácio dióxido de carbono – calcário	[kg]
r_t	Parâmetro de tempo	[s]
Re_i	Número de Reynolds	[-]
$R_{conv_{ex}}$	Resistência térmica parede exterior – ambiente	[m ² K/W]
$R_{conv_{int}}$	Resistência térmica inteiro – parede inteiro	[m ² K/W]
$R_{cond_{ref}}$	Resistência térmica do tijolo refratário	[m ² K/W]
$R_{cond_{iso}}$	Resistência térmica do isolamento	[m ² K/W]
t	Tempo de residência	[h]
T_{amb}	Temperatura do ar ambiente	[°C]
$T_{g_{1-2}}$	Temperatura do fluido (ar) à entrada da zona de calcinação	[°C]
$T_{g_{2-3}}$	Temperatura dos gases à saída da zona de calcinação	[°C]
T_{g_e}	Temperatura do fluido (ar) à entrada da zona de arrefecimento	[°C]
T_{gs}	Temperatura dos gases de escape	[°C]
T_{int}	Temperatura na zona de calcinação	[°C]
T_{par}	Temperatura da parede exterior do forno	[°C]
T_{pe}	Temperatura da partícula à entrada do forno (zona de pré-aquecimento)	[°C]
$T_{p_{2-1}}$	Temperatura da partícula à saída da zona de calcinação	[°C]
$T_{p_{3-2}}$	Temperatura da partícula à entrada da zona de calcinação	[°C]
T_{ps}	Temperatura da partícula à saída da zona de arrefecimento	[°C]
U_{fi}	Velocidade do fluido na zona i	[m/s]
V_{liq_t}	Volume líquido do forno	[m ³]
V_{liq_c}	Volume líquido da zona de calcinação	[m ³]
V_p	Volume de uma partícula (pedra)	[m ³]
V	Volume total do forno	[m ³]
V_c	Volume da zona de calcinação	[m ³]
X_i	Fração molar	[-]
Y_i	Fração mássica	[-]

Letras gregas

α	Majoração para perda de carga (fator de segurança)	[-]
β	Coeficiente de expansão volumétrica	[K ⁻¹]
Δp_i	Perda de carga na zona i	[Pa]
ε	Porosidade	[-]
ϵ	Emissibilidade	[-]
θ_{nec}	Energia necessária para produzir uma quantidade de CaO	[kJ]
$\theta_{reação}$	Energia necessário no processo de decomposição do calcário	[kJ/kg CaO]
ϑ	Área de transferência por unidade de volume	[m ² /m ³]
μ	Viscosidade dinâmica do gás de combustão	[N s/m ²]
μ_i	Viscosidade dinâmica do fluido (ar) na zona i	[N s/m ²]
ν	Viscosidade cinemática do fluido (ar)	[m ² /s]
ρ_{CaCO_3}	Massa específica do CaCO ₃	[kg/m ³]
ρ_{g_i}	Massa específica do fluido	[kg/m ³]
ρ_{GN}	Massa específica do Gás Natural	[kg/m ³]
σ	Constante de Stefan-Boltzmann	[W/(m ² K ⁴)]
ϕ_{ij}	Coeficiente de Wilke	[-]

Índices

i	Zonas do forno (1=arrefecimento, 2=calcinação, 3=pré-aquecimento)
amb	Ambiente exterior ao forno
g	Fluido gasoso
c	Zona de calcinação
int	Ambiente no interior do forno
p	Partícula
par	Parede do forno

Siglas

GN	Gás Natural
----	-------------

Capítulo 1

Introdução

A cal, o principal produto da calcinação, é um produto muito versátil que se destaca em todo o mercado mundial, a sua produção ocorre através de fornos calcinatórios, os quais se encontram com diversas configurações construtivas e com diferentes produtividades. Ela está presente em quase todos os setores produtivos, desde a construção civil, tratamento de águas, agricultura, alimentação e infinitas outras aplicações.

A calcinação do calcário, que apresenta como principal componente o óxido de cálcio, ocorre dentro de fornos específicos de calcinação a temperaturas elevadas (entre 900 e 1200°C).

Os fornos verticais são os que mais se destacam na indústria de produção de cal e são a tecnologia mais antiga, adequando-se normalmente às exigências do mercado, nomeadamente as respeitantes à qualidade da cal. Os fornos são revestidos internamente por tijolos refratários que protegem as placas metálicas do forno e mantêm a temperatura interna dentro dos valores necessários ao processo tecnológico em questão.

1.1. – Motivação

O forno vertical é vastamente utilizado na indústria. Apesar de ter sido usado na indústria desde séculos, tem havido poucos trabalhos de pesquisa sobre este tipo de fornos.

Os produtores de cal partilham uma preocupação quanto à quantidade e qualidade do produto final. Uma vez que a qualidade e as características da cal varia consideravelmente, os produtores muitas vezes utilizam testes químicos e físicos para garantir as exigências do produto final. A maior parte destes testes são muito demorados e bastantes dispendiosos. A fim de reduzir custos desde o início da produção os produtores devem tomar todas as hipóteses, por isso a utilização de um programa de simulação para satisfazer as suas necessidades, antes ou na fase de projeto de um novo forno é fundamental.

Uma alternativa viável a um complexo programa de simulação é o desenvolvimento de algoritmos de cálculo pelos próprios construtores ou utilizadores.

1.2. – Objetivo

A monografia apresentada tem como principal objetivo o desenvolvimento de um programa de computador para o dimensionamento de fornos verticais de cal. O programa de simulação foi desenvolvido como uma ferramenta de auxílio ao projeto térmico do forno vertical tradicional, permitindo avaliar o seu comportamento térmico e capacidade de produção, com vista a um aumento da qualidade do processo de calcinação. A utilização desta ferramenta na fase de projeto de um forno permite a redução das incertezas de projeto e uma melhor avaliação da sua viabilidade técnica e económica, ao permitir a fácil obtenção de uma série de dimensões e potências de operação, que permitam ao projetista uma escolha expedita da solução mais adequada.

1.3. – Estrutura da dissertação

A presente dissertação encontra-se dividida em 8 capítulos, conforme se descreve sinteticamente de seguida.

O Capítulo 1 introduz o tema, descrevem-se os objetivos e motivações da presente dissertação. Apresenta-se o resumo de cada um dos capítulos da dissertação. No Capítulo 2 é apresentada uma revisão bibliográfica sobre os principais temas desta dissertação, nomeadamente, a cal. Assim, é exibida uma abordagem à sua história, origem mineral, componentes e aplicações. De seguida, faz-se uma breve descrição das técnicas de produção de cal, como extração, desmonte e fornos de calcinação. No Capítulo 3 é apresentado o tipo de forno projetado nesta tese, tanto a sua vertente geométrica, como no que se refere à sua adaptação a um modelo numérico. No Capítulo 4 prossegue-se com o processo de calcinação, onde são expostos vários aspetos relativos ao processo. Inicialmente é apresentado uma pequena introdução, a degradação térmica. O formato e distribuição do tamanho da partícula é também tido em consideração. Finalmente, são destacados os aspetos referentes à cinética de calcinação e fatores que a influenciam. No Capítulo 5 são descritos, de forma genérica, os modelos matemáticos utilizados na simulação, sendo apresentadas as formas básicas das equações, assim como o modelo escolhido para a combustão. O processo de cálculo iterativo é descrito de forma simplificada. No Capítulo 6 é apresentado, de uma forma pormenorizada, o desenvolvimento do programa, as suas funcionalidades e por fim o seu aspeto gráfico. No Capítulo 7 são apresentados os resultados obtidos da simulação numérica e finalmente no Capítulo 8 apresentam-se as conclusões resultantes da realização desta dissertação, bem como sugestões para um trabalho futuro a realizar nesta área.

Esta tese contém ainda Anexos, onde se encontram informações que foram consideradas como secundárias para o corpo principal da dissertação, mas que servem de complemento ao trabalho realizado.

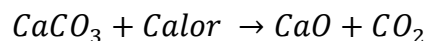
Capítulo 2

A cal

Neste capítulo são apresentados os fundamentos e alguns dos principais trabalhos publicados na literatura, relacionados com este estudo. Inicialmente são introduzidos conceitos relativos à produção de óxido de cálcio, os seus aspetos básicos, a sua importância, as suas aplicações e, consequentemente, as suas técnicas de produção.

2.1. – Introdução

A cal é o produto que resulta da decomposição térmica do calcário ou carbonato de cálcio. A reação química que lhe dá origem é simples, mas requer recursos e conhecimentos complexos para a sua perfeita realização.



Esta reação endotérmica dá-se a temperaturas na ordem dos 900 °C consumindo 3029 kJ por kg de CaO produzido.

A produção de cal é uma indústria global que contribui muito para o desenvolvimento social e económico em todo o mundo. Muitas aplicações industriais e de consumo benéfico tornaram-se possíveis através da utilização de cal (Boynton, 1980).

2.2. – História

A história das civilizações mostra que há uma ligação íntima entre os estágios de desenvolvimento alcançados pelo Homem e o emprego dos minerais nas suas atividades. Não se sabe ao certo quando e como o homem se apossou dos conhecimentos sobre a técnica ou arte de calcinar as rochas calcárias para obter o “pó-branco” que aguçou a sua curiosidade e o levou a aplicá-lo nas pinturas rupestres e em matérias cimentares.

A produção de cal e as suas aplicações podem ser rastreadas até à civilização Romana, Grega e Egípcia, mas a primeira informação definitiva escrita relativamente à cal foi proferida pelos Romanos. No livro “*De Architectura*”, *Marcus Pollio*, um engenheiro e arquiteto famoso que viveu durante o reinado de Augusto (27 A.C a 14 D.C.), apresenta muitas técnicas do uso de cal para argamassas envolvidas na construção de obras portuárias, pavimentos/calçados e prédios. Na nova colónia Americana, a queima bruta de calcário foi um dos processos de fabricação iniciais desenvolvidas pelos colonos, usando fornos “*dug-out*” construídos de tijolo comum ou alvenaria na encosta de uma colina, com carvão ou uma fogueira na parte inferior e um tempo de queima de 72 h. Estes ainda podem ser vistos em zonas mais antigas do país. Nos últimos anos, sob a influência de investigações na área da engenharia química, a fabricação de cal tornou-se uma grande indústria sob um exato controlo técnico, com a uniformidade resultante de produtos, e com um custo menor (Shreve, 1977).

No ano de 1217, a cal (virgem e moída) é pela primeira vez utilizada em guerra. Os ingleses lançavam-na contra os rostos dos seus inimigos franceses (Boynton, 1966). Em contrapartida, durante o período da Renascença, em torno de 1500 na Itália, célebres artistas, como Miguel Ângelo e Rafael, utilizavam com frequência uma pasta branca à base de cal nos seus murais e frescos (Guimarães, 2002). Em 1775, o químico inglês Joseph Black, em Edimburgo, dá a primeira explicação técnico-científica sobre a calcinação de calcários, incluindo a expulsão do anidrido carbónico como gás. Poucos anos mais tarde, o célebre químico Lavoisier reafirmou a teoria de Black (Boynton, 1966). No ano de 1780, Higgins publica os resultados das suas pesquisas demonstrando que o endurecimento da cal resulta da sua combinação com o gás carbónico do ar (Leprevost, 1953).

Com a afirmação dos princípios científicos básicos relativos à fabricação de cal, iniciou-se no final do século XIX um novo período de evolução industrial

do produto – mineração, tratamento do minério, fornos – combustíveis, energia, economia e recursos humanos. Essa evolução é revelada não só pelo crescimento contínuo da produção como também pela multiplicidade de aplicações alcançada pelo produto nos dias atuais.

2.3. – Origem mineral

A parte sólida da crosta terrestre, aproximadamente 21% do planeta, é constituída por massas sólidas, dispostas em relevos dos mais variados. Essas massas, denominadas rochas, são constituídas basicamente por compostos químicos cristalograficamente organizados, minerais. Formaram-se desde a época pré-cambriana, entre 4 bilhões e 4,5 bilhões de anos atrás, e nas eras seguintes até aos dias atuais. Dispuseram-se como consequência de fenômenos químicos e físicos ocorridos na sua história geológica (Wikipédia, 2015).

Entre as rochas e minerais utilizados pelo homem, destacam-se as rochas carbonatadas, que constituem 0,25% do volume da crosta. Encontram-se encaixadas não só nas formações mais antigas – arqueanas – como nas mais modernas, cenozoicas. Em geral, são sedimentos, metamorfizados ou não, e mais raramente são rochas intrusivas relacionadas à atividade vulcânica alcalina.

Essas rochas são constituídas em 50% ou mais pelos minerais:

- a. Calcita: carbonato de cálcio (CaCO_3) romboédrico;
- b. Aragonita: carbonato de cálcio (CaCO_3) ortorrômbico;
- c. Dolomita: carbonato de cálcio e carbonato de magnésio (CaCO_3 . MgCO_3) romboédrico.

Entre os componentes minerais considerados como impurezas ou como secundários estão: quartzo, certos silicatos argilosos, óxidos metálicos de ferro e manganês, matéria orgânica, fosfatos, sulfeto, sulfatos, fluoretos e brucita. Os elementos-traços que acompanham mais frequentemente a calcita e a dolomita são: alumínio, titânio, sódio, potássio, bário, estrôncio, chumbo, cádmio, vanádio, cromo, cobalto, níquel, cobre, gálio, zircônio, molibdênio e estanho.

Entre as várias propostas de classificação das rochas carbonatadas cálcio-magnesianas a mais aceite na literatura por pesquisadores da área está descrita na Tabela 2.1 (Guimarães, 2002).

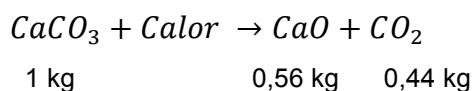
Tabela 2.1 - Classificação das rochas carbonatadas cálcio-magnesianas (Guimarães, 2002).

Minério	CaCO ₃ / MgCO ₃	Ca / Mg
Calcário Calcítico	> 100	> 105
Calcário Magnesiano	10 – 3,5	105 – 60
Calcário Dolomítico	3,5 – 1,5	60 – 16
Dolomito	1,5 – 1,2	-

As reservas de calcários/dolomitos são disseminadas pela maior parte dos continentes. Em Portugal, as reservas de pedra calcária localizam-se no Maciço Estremenho da Serra dos Candeeiros no Concelho de Santarém, nos Maciços Calcários de Souselas e de Brasfemes a cerca de 10 km de Coimbra. Existe ainda a Pedreira de Calcário da região de S. Brás de Alportel, no Algarve, e outros afloramentos de menor expressão (LNEG, 2015).

2.4. – Componentes da cal

A produção de cal envolve a calcinação de calcário, libertando grande quantidade de CO₂ na atmosfera, a equação dessa calcinação fica:



Isso significa que, para cada tonelada de cal, são produzidos 785 kg de CO₂. A obtenção de cal ocorre em fornos que transformam a calcita (carbonato de cálcio – CaCO₃) o calcário calcítico em cal (óxido de cálcio – CaO).

2.4.1. – Características físico-químicas

Quando provém de rochas carbonatadas puras, a cal é um produto inorgânico branco. Quando apresenta colorações bege, amarelada e levemente cinza, é sinal que detém impurezas. O seu odor é equivalente ao terroso. Forma-se com estrutura cristalina, em cristais isolados ou conglomerados cristalinos de dimensões e espaços intercristalinos variados observáveis somente pela microscopia. Por vezes, a sua observação é difícil, pois as suas configurações aparecem compactas ou microcristalinas. O seu cristal pertence ao sistema cúbico, com lados de 4,797 ângströms no óxido de cálcio e 4,203 ângströms no óxido de magnésio (Guimarães, 2002).

A Tabela 2.2 apresenta as principais características físicas e químicas da cal.

Tabela 2.2 - Características físicas e químicas da cal (Guimarães, 2002).

Característica	CaO
Densidade	3,2 a 3,4
Massa volúmica aparente	811 a 963 kg/m ³
Ângulo de repouso	55 °
Ca presente	71,47 %
Massa molecular	56 kg/kmol
Dureza	2 – 4 MOHS
Ponto de fusão	2570 °C
Calor de formação	151,9 kcal/mol
Porosidade	18 a 48 % (média – 35%)

2.5. – Aplicações da cal

A cal recebe dos técnicos a denominação de material versátil e social, pela multiplicidade de aplicações e pela contribuição ao bem-estar das comunidades. Apesar do volume da sua produção ser inferior aos líderes mundiais da produção de origem mineral (petróleo, carvão), a cal destaca-se pelo seu consumo misto, seja como insumo, seja como produto. Em levantamentos relativamente recentes a cal inclui-se entre os 10 primeiros nas estatísticas de origem mineral.

A cal é o alcalino mais barato, e o mais utilizado, pois tem uma aplicação vasta na agricultura e na indústria:

- Agricultura: A cal aumenta a eficiência do fertilizante. A calagem de um solo ácido aumenta o pH do solo, os níveis de cálcio e magnésio, acelera a atividade microbológica e aumenta a taxa de libertação de matéria orgânica e elementos nutritivos do solo;
- Indústria do alumínio: A cal ajuda a remover a sílica da bauxite durante o fabrico de óxido de alumínio;
- Construção: A cal é usada para o fabrico de tijolo sílica-cal, isolamento e matérias de construção. É usada na argamassa para fixar os tijolos de parede. Em conjunto com o betão e o gesso aumenta o desempenho destes;
- Indústria da alimentação: A cal reage com o suco de açúcar bruto para a produção de açúcares de cana e de beterraba. É também um

ingrediente no bicarbonato de sódio e ajuda a preservar a frescura de frutas e vegetais;

- e) Indústria de tratamento de água residuais: A cal neutraliza resíduos ácidos gerados na indústria geral impedindo assim a corrosão e protegendo o meio ambiente. Também remove a sílica, o manganês, os fluoretos, e o ferro entre outras impurezas na água;
- f) Indústria de extração de metais: A cal serve como forma de incrementador da fluutuabilidade na recuperação de cobre, mercúrio, zinco, níquel, cumbo, ouro e prata;
- g) Fabrico de papel: Os fabricantes de celulose e papel usam cal para recuperar hidróxido de sódio durante a conversão das aparas de madeira para celulose. A cal é um branqueador ou alvejante da celulose e dissolve também os componentes não-celulósicos da palha e desintegra as fibras durante o fabrico de papelão e cartão;
- h) Controlo da poluição: A cal é usada para absorver dióxido de enxofre dos gases de escape de fundições e centrais termoelétricas;
- i) Construção de estradas: A cal converte a argila instável em subclasses para eliminar a formação de torrões. Cria um solo que não se deforma (não encolhe nem dilata). Pode fornecer uma ação de cimentação que estabiliza o solo numa camada firme impermeável à penetração da água;
- j) Tratamento de esgotos: A cal reduz a poluição ao remover matéria orgânica, fosfatos e compostos azotados provenientes das águas residuais. Evita o excesso de vegetação nos lagos, controla odores de estações de tratamento de água residuais;
- k) Estabilização de solos: A cal pode ser utilizada para estabilizar solos encharcados;
- l) Indústria metalúrgica: A cal é utilizada como fluxo de purificação do aço e para remover impurezas de fósforo, enxofre e silício. Lubrifica as hastes de aço à medida que estas são puxadas através de matrizes para formar fio. Como um revestimento, para impedir lingotes nos moldes durante a fundição de ferro. Um banho de cal neutraliza os traços de decapagem aderente aos produtos de aço;
- m) Indústria dos curtumes: Remove pelos e peles gordas na preparação para a curtimenta do couro;
- n) Tratamento da água: A cal trata de água industrial potável e de abastecimento, incluindo água potável para residências e água para

processos industriais. Suaviza a água removendo a dureza, os bicarbonatos e desinfeta a água contra as bactérias.

As suas aplicações chegam quase a uma centena, conforme a Tabela 2.3.

Tabela 2.3 - Aplicações da cal (Boynton,1980).

APLICAÇÕES DA CAL	
Agentes de Processos Químicos e Físico-químico	Sector de Consumo
Absorção	Branqueamento Remoção do SO ₂ , SO ₃ Processo sulfito (fabricação de papel) Armazenamento de frutas
Matéria-prima	Borracha Cimento Alimentos Cianamida cálcica Álcalis Tintas Carbureto de cálcio Inseticidas Abrasivos Vidro
Desidratação	Secagem de ar Borracha Solventes orgânicos Álcool
Floculação	Açúcar Flotação de minérios Tratamento de águas residuais Tratamento de água para fins potáveis Tratamento de esgotos Pigmentos de tintas
Fluxo	Fornos de aço LD – BOF Fornos de aço <i>Martin-Simens</i> Fornos de aço elétricos Sinterização Metais não ferrosos

Lubrificação	Lama de sondagens
	Trefilação de arames
Aglomeração	Argamassa de assentamento
	Reboco e emboço
	Misturas asfálticas
	Matérias isolantes
	Misturas solo-cal
	Produto com silicato cálcio
	Tijolo silico-cal
	Peletização de minério de ferro
Neutralização	Estuques
	Ácido cítrico
	Tratamento de águas
	Fertilizantes
	Resíduos de decapagem de metais
	Resíduos de explosivos
	Laticínios
	Drenagem de águas de minas
	Resíduos radioativos
	Resíduos de urânio
	Calagem
	Resíduos de cromo
	Resíduos de corantes
Solução	Gelatinas
	Couro (despelador)
	Tintas à base de caseína
	Papelão
Causticação	Recuperação de hidróxido de sódio
	Processo de sulfato e soda (fabricação de papel)
	Lavagem alcalina
Hidrolisar	Produtos de celulose
	Graxa lubrificante
	Compostos derivados de cloro
	Curtumes

2.6. – Técnicas de produção

Após a extração da jazida, a pedra calcária é transportada até à fábrica para ser submetida a um processo de britagem. De seguida, pode ser moída e embalada para comercialização, ou pode levar-se até ao forno de calcinação para ser transformada. Após a calcinação, a cal pode ser moída e comercializada na forma de cal, ou pode ainda sofrer um processo de hidratação e ser finalmente comercializada (Figura 2.1).

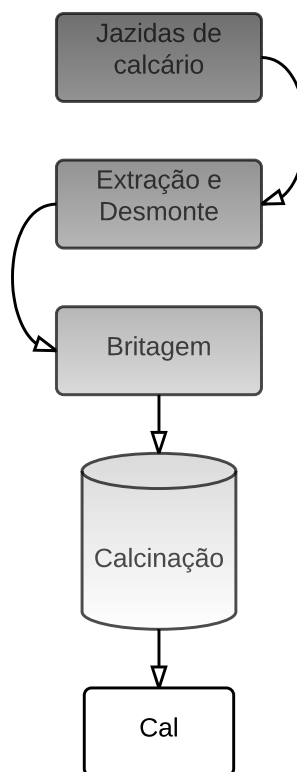


Figura 2.1 - Sequência de fabrico da cal.

2.6.1 – Extração, desmonte e britagem

A extração da rocha calcária pode ser feita a céu aberto ou em galeria, através da exploração da jazida de calcário. Para que a extração do calcário seja de alta qualidade, há que efetuar primeiramente um rigoroso processo de seleção do calcário e uma limpeza total da rocha, para que se alcancem as características físicas e químicas necessárias às diversas aplicações da cal. Depois de limpa, a rocha é perfurada e o pé expelido é recolhido para a realização das análises em laboratório. Verificada a qualidade do material, faz-

se o desmonte da rocha, por meio de detonação com recurso a explosivos. (Figura 2.2)

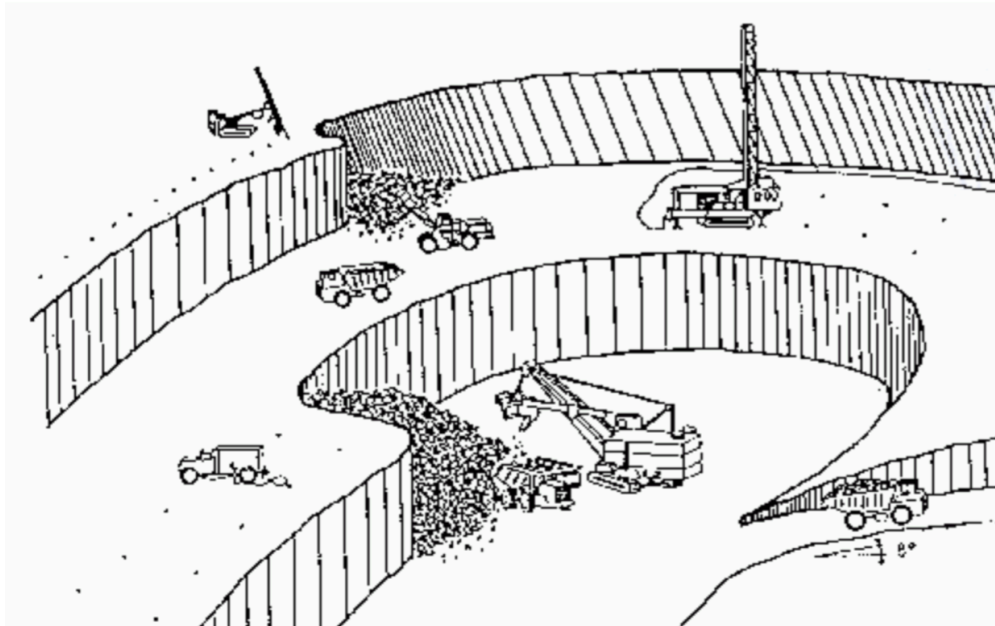


Figura 2.2 - Operações de extração e desmonte (LNEG, 2015)

A rocha desmontada é levada, de seguida, para uma britagem primária. A britagem é um processo de fragmentação da pedra calcária. É realizada em britadores, onde é feita a seleção, por meio de peneiros vibratórios, da pedra destinada aos diversos produtos. Esta etapa tem por objetivo retirar as impurezas do processo e classifica o calcário por diferentes especificações granulométricas, até a obtenção da bitola de calcário apropriada ao processo de calcinação. Este processo pode ser dividido em várias etapas de britagem e estas etapas são diferenciadas pela sua capacidade de redução das pedras para a classificação em peneiras dimensionadas para cada produto final. Após a pré-moagem nos britadores, o calcário segue para a segunda fase de moagem em equipamentos denominados de moinhos. Estes proporcionam uma moagem mais precisa, ou seja, o produto sai do moinho com uma granulometria mais uniforme. Os finos que não são utilizados no processo de calcinação são destinados à venda como agregados e também para a produção de calcário agrícola.

As correias transportadoras são responsáveis por fazer o transporte do material da zona de britagem até aos fornos de calcinação.

2.6.2. – Fornos de calcinação

A calcinação dos calcários pode ser levada a cabo em fornos contínuos, sendo que estes podem ser verticais ou horizontais.

2.6.2.1. – Fornos industriais

À época da construção das pirâmides egípcias, utilizando-se lenha e depois o carvão como combustível, certamente os meios de produção do conhecido “pó-branco” dos calcários elevou-se do solo para grandes piras com 1,5 a 2 metros de altura e igual diâmetro. No interior das piras, troncos e galhos de árvores eram habitualmente entremeados com carvão mineral, dispostos com leitos de pedra calcária e depois incendiados, a dosagem dos combustíveis era feita pela prática. O fogo era iniciado com ajuda de um tronco de madeira colocado entre os componentes da carga da pira e durava até o consumo total do combustível, quando restavam apenas finas cinzas e blocos de “pó-branco”. Este sistema primitivo de calcinação atravessou muitos séculos (Guimarães, 2002).

Em centros de desenvolvimento houve um grande progresso dos fornos, que passaram por inovações relevantes. Transformaram-se em estruturas de grande porte, encravadas ou não na meia encosta dos terrenos; ganharam revestimentos, chaminés para melhorar a tiragem; aparelhos para tiragem forçada; interdependência entre os focos de calor, de descarga de cal e de cinzas; e novos combustíveis passaram a ser utilizados. Toda essa evolução dos modelos de fornos foi dirigida no sentido da economia de combustível, da eficiência térmica e da qualidade de cal (Guimarães, 2002).

2.6.2.2. – Fornos de barranco

Os fornos de barranco, designados assim por inicialmente se apoiarem em encostas, são geralmente intermitentes e de forma troncocónica, com 4 ou 5 metros de altura, abertos, e as chamas a chegarem às proximidades do teto. Estes são grandes devoradores de combustível, tanto quando este é misturado com a carga e, principalmente, quando queimado em fornalhas laterais. Em alguns casos, os fornos intermitentes não eram apoiados nos barrancos, como os tipo “pote”. Eram construídos isolados numa plataforma (Guimarães, 2002). Desta forma, a carga com o diâmetro superior a 150 mm podia ser empilhada manualmente no interior do forno ou repousar sobre uma abóbada, também feita com trabalho manual e com pedras devidamente escolhidas. No primeiro caso, este possui aberturas frontais que funcionam como fornalha e depois da

queima servem para descarga. No outro, a abóbada serve como teto de uma fornalha com um pequena abertura para introdução do combustível. Ambos os tipos possuem uma cavidade inferior para recolher as cinzas. O seu altíssimo consumo de combustível, variável entre largos limites, pode alcançar o equivalente a 5 m³ de lenha por tonelada de cal (Guimarães, 2002).

A Figura 2.3 apresenta um forno de barranco em cuba ovoide de alvenaria, com dispositivo para impedir a penetração de água das chuvas e forçar a passagem dos gases pelos canais laterais de tiragem.

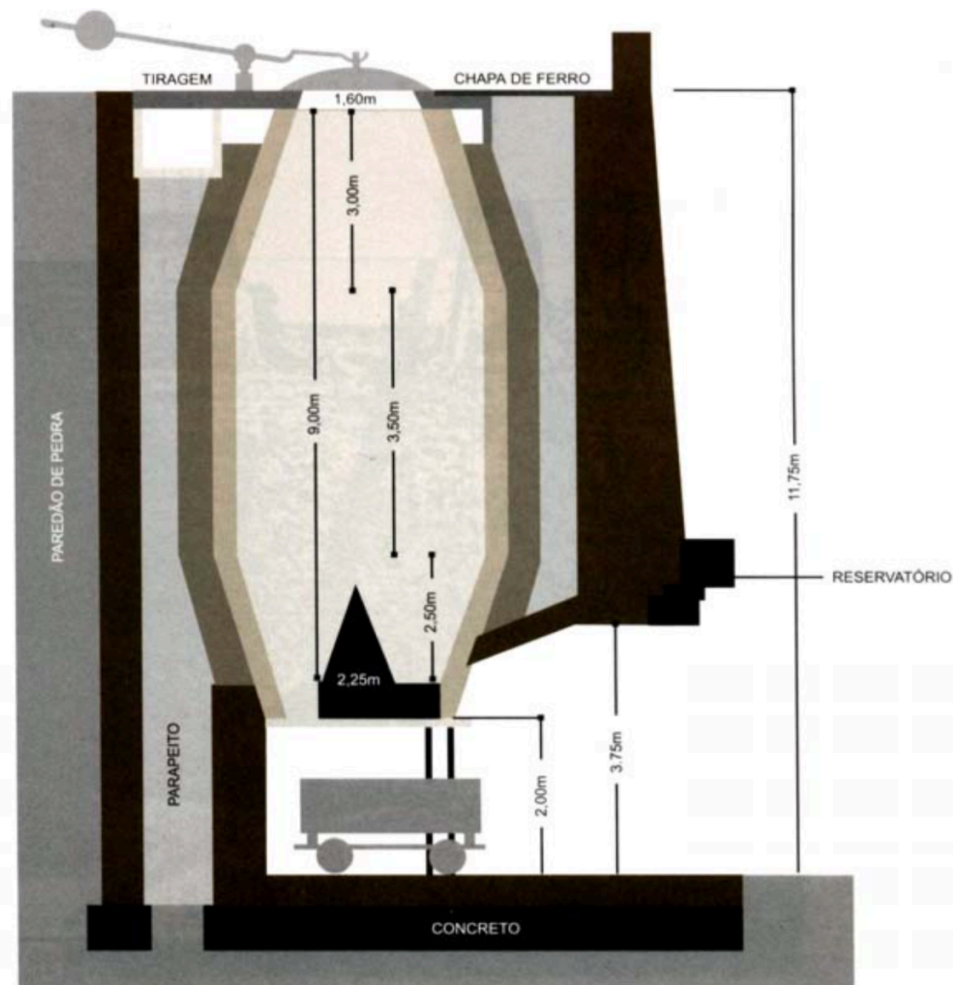


Figura 2.3 - Forno de barranco em cuba ovoide de alvenaria (Guimarães, 2002).

2.6.2.3. – Fornos verticais

Enquanto em muitas regiões do mundo os fornos descontínuos ainda são responsáveis pela produção de cal, em alguns países europeus, sul-americanos e norte-americanos a calcinação de calcários passou a ser feita em fornos de características estruturais e térmicas mais aperfeiçoadas. Os fornos ganharam mais altura, forma cilíndrica ou poligonal, revestimento de tijolos refratários, três ou mais zonas de queima, cinzeiro, carga e descarga semiautomática, funcionando em produção contínua. Os modelos de fornos contínuos de alvenaria, bem como os seus sistemas de queima (tiragem, combustíveis, carga, descarga, isolamento térmico) foram melhorados desde o fim do século XIX em todo o mundo, até culminarem com os modernos fornos de calcinação, já com estrutura metálica e zonas definidas de pré-aquecimento, calcinação e arrefecimento.

Na Figura 2.4, exibem-se modelos de fornos verticais contínuos, de alvenaria ou revestidos com chapas de aço, publicados em *Theory and Practice of Lime Manufacture* por Lowel Andrew e N, C, Rockwood, em 1946 na *American Society of Mechanical Engineers*. (Guimarães, 2002).

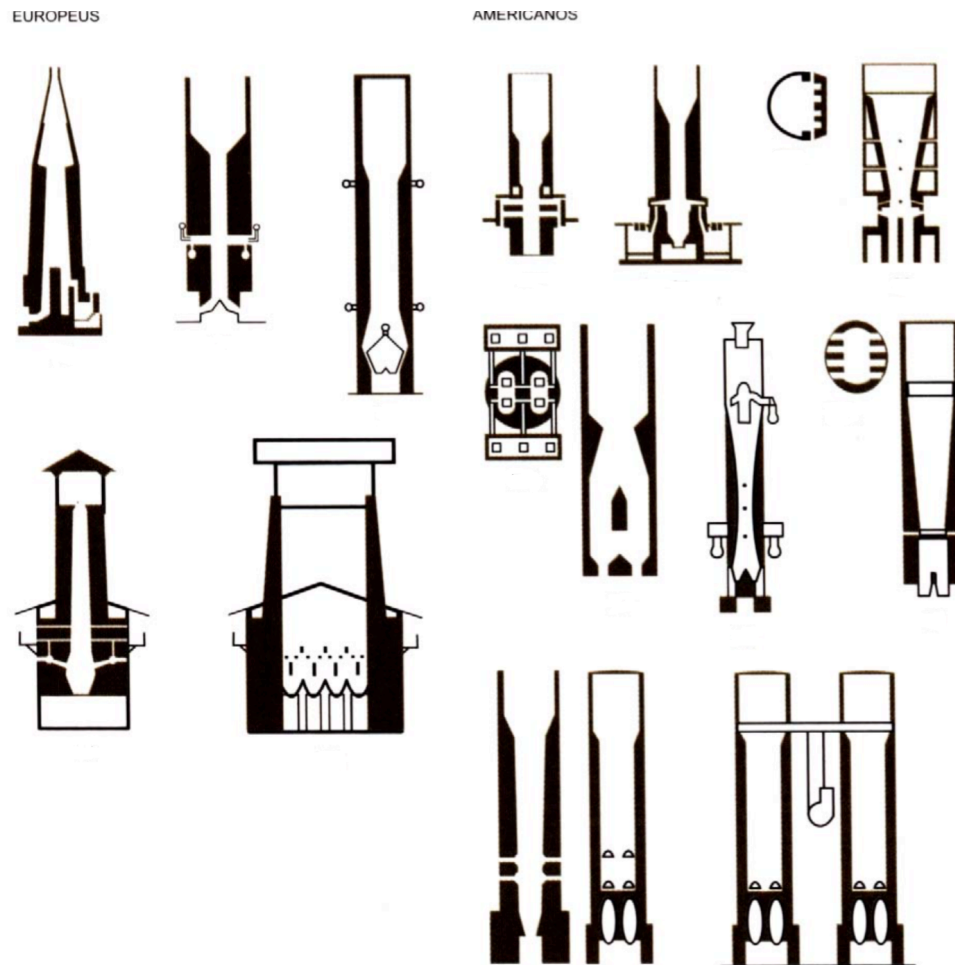


Figura 2.4 - Tipo de fornos contínuos verticais para a produção de cal (Guimarães, 2002).

A modernização da produção de cal começou com o aparecimento de instalações com fornos verticais metálicos contínuos, com as seguintes características: tiragem forçada, formas cilíndricas, camisas de refratários, carga e descarga automática, zonas de armazenamento / pré-aquecimento / calcinação / arrefecimento da cal (que permitem a recuperação de calor perdido nos gases e no calcinado) e capacidade de produção de até 700 toneladas por dia (Perona, 1976). A alimentação destes fornos mais modernos é feita com minério, geralmente de 50 a 100 mm. As Figura 2.5 e Figura 2.6 apresentam um esquema do forno vertical da marca *AZBE*. A Figura 2.7 exhibe um esquema de outro forno da marca *West*, sendo apresentadas algumas características do mesmo na Tabela 2.4 (Perona, 1976).

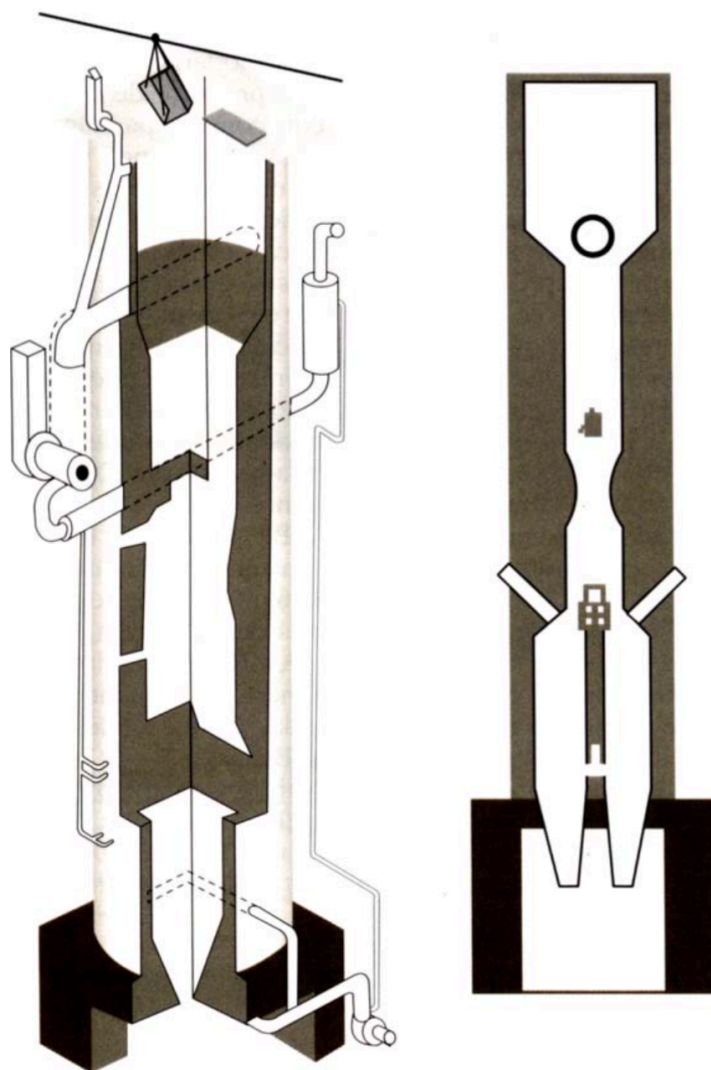


Figura 2.5 - Forno de calcinação AZBE (Perona, 1976).

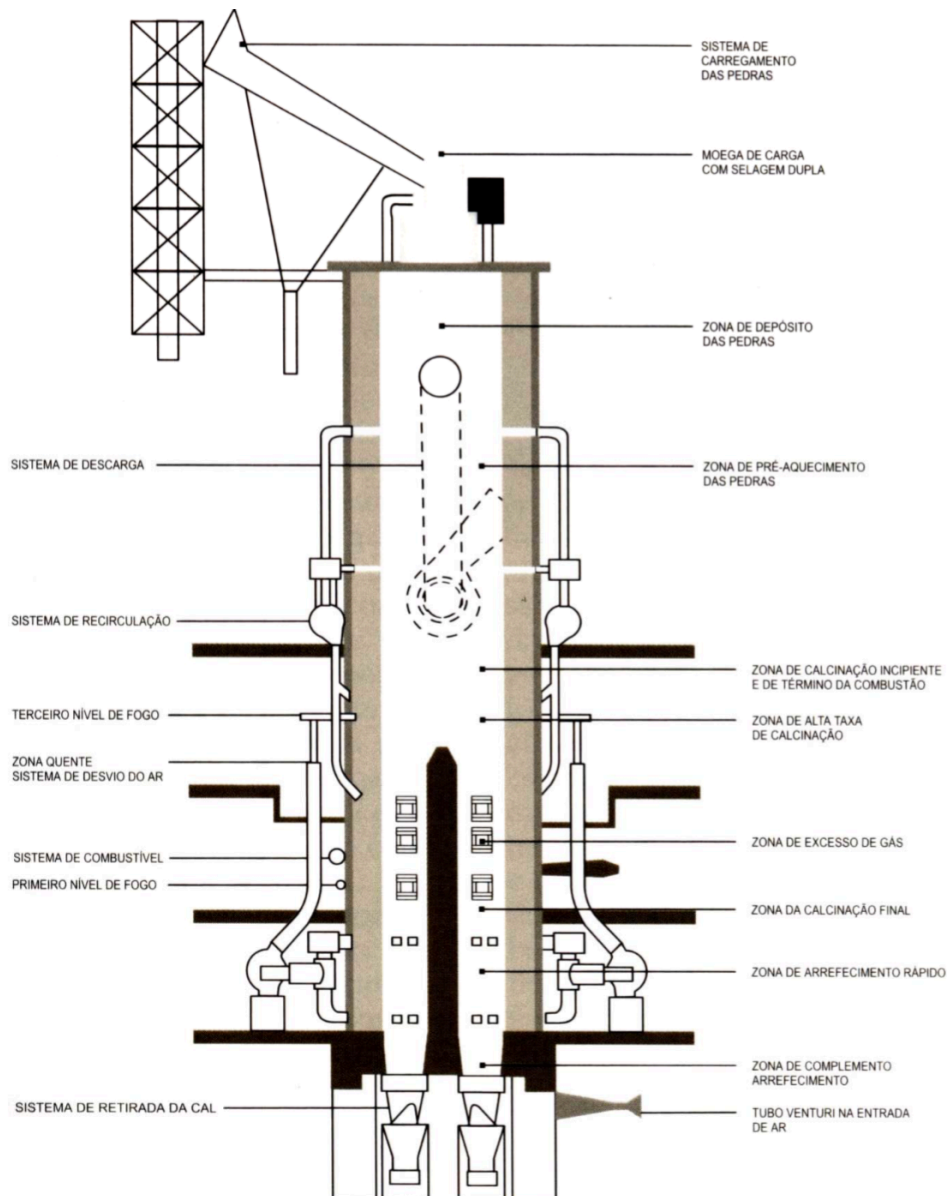


Figura 2.6 - Forno de calcinação AZBE a gás natural, com controlo integral de temperatura na zona de calcinação (Perona, 1976).

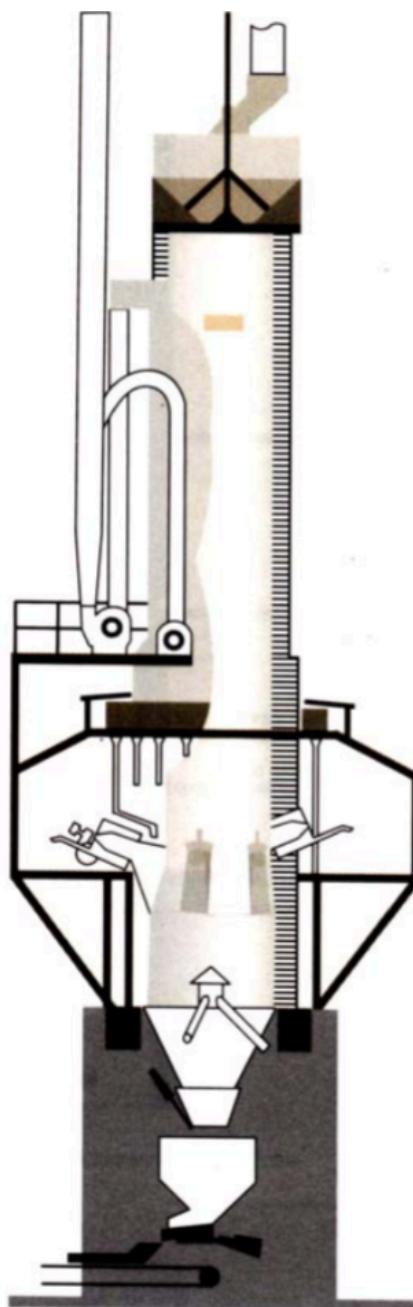


Figura 2.7 - Forno de calcinação West. $\text{CO}_2 < 2\%$; consumo de 4605,5 kJ/kg; consumo energético de 15 kWh/t de cal; flexibilidade de capacidade de 60% (Perona, 1976).

Tabela 2.4 - Características de produção de um forno de calcinação West (Perona, 1976).

Produção (t/dia)	Granulometria Típica	
30	60 – 90 mm	50 – 75 mm
50	65 – 90 mm	90 – 130 mm
70	90 – 130 mm	100 – 150 mm
90	90 – 130 mm	100 – 150 mm
120	90 – 130 mm	100 – 150 mm
150	100 – 150 mm	125 – 175 mm
200	100 – 150 mm	125 – 175 mm

A procura da economia de energia e de melhor reatividade para a cal conduziu ao aparecimento dos fornos de calcinação verticais de cubas múltiplas e escoamentos paralelos alimentados a óleo combustível. São equipamentos de 2 ou 3 cubas operando em paralelo. No sistema de produção, um deles está sempre em funcionamento na vertical descendente de calcinação, enquanto os outros (ou o outro) operam em contracorrente ascendente como trocadores de calor.

Após cada período de recuperação (10 a 15 minutos), o escoamento do gás é revertido e a cuba anteriormente usada com escoamento paralelo passa a operar com escoamento em contracorrente. Tal reversão permite que a parte superior da cuba funcione, alternadamente, como recuperador do calor do ar primário ou como arrefecedor dos gases de combustão. Na cuba calcinadora a temperatura normal de operação é de aproximadamente 950 a 1050 °C (Guimarães, 2002). Como consequência, as temperaturas dos gases da chaminé são extremamente baixas, verificando-se igualmente com o consumo de combustível do sistema (3559 a 3797 kJ/kg de cal). A granulometria da carga obedece à relação 1:2 até 1:4. Os modelos mais presentes na indústria da cal são os da *Maerz Ofenbau A. G.*, de Zurich, Suíça, e o de sua licenciada *Kennedy Van Saun Corporation*, Danville, Pa, U.S.A., com capacidade raramente excedendo 600 t/dia. A escala de granulometria obedecida nas unidades pequenas é de 25,4 x 63,5 mm e 69,85 x 139,7 mm nos modelos maiores (Guimarães, 2002).

Nos modelos de múltiplas câmaras (500 a 1000 t/dia), denominados *Flash* e *Calciner*, que têm a vantagem de injetar finos de calcário (3,175 mm), as condutas possuem uma grande superfície interior que permite velocidade suficiente para transportar as partículas para os ciclones. Daí as partículas são separadas na corrente gasosa e alimentam outro sistema duto/ciclone para pré-aquecimento ou alimentação à calcinação em suspensão. Devido à grande área superficial das partículas e à boa mistura com os gases quentes, a transferência de calor é quase ideal. Uma série de condutas é usada para aquecer o calcário antes de sua entrada no calcinador, onde o calor do gás é extraído. O seu consumo de combustível é baixo, da ordem de 4605,5 kJ/kg de cal, o que permite utilizar com baixo custo os finos de calcário das usinas tradicionais de fornos verticais e rotativos (Guimarães, 2002).

A Figura 2.8 apresenta uma vista geral do forno de calcinação *Maerz Ofenbau (Parallel Shaft)*. A Figura 2.9 exibe um esquema de funcionamento de fornos *Parallel Shaft* da *Maerz Ofenbau*, o qual consiste de três fases: uma fase de combustão na cuba A, uma fase de alimentação de calcário e descarga de cal, e finalmente uma fase de combustão na cuba B, repetindo-se o ciclo periodicamente.

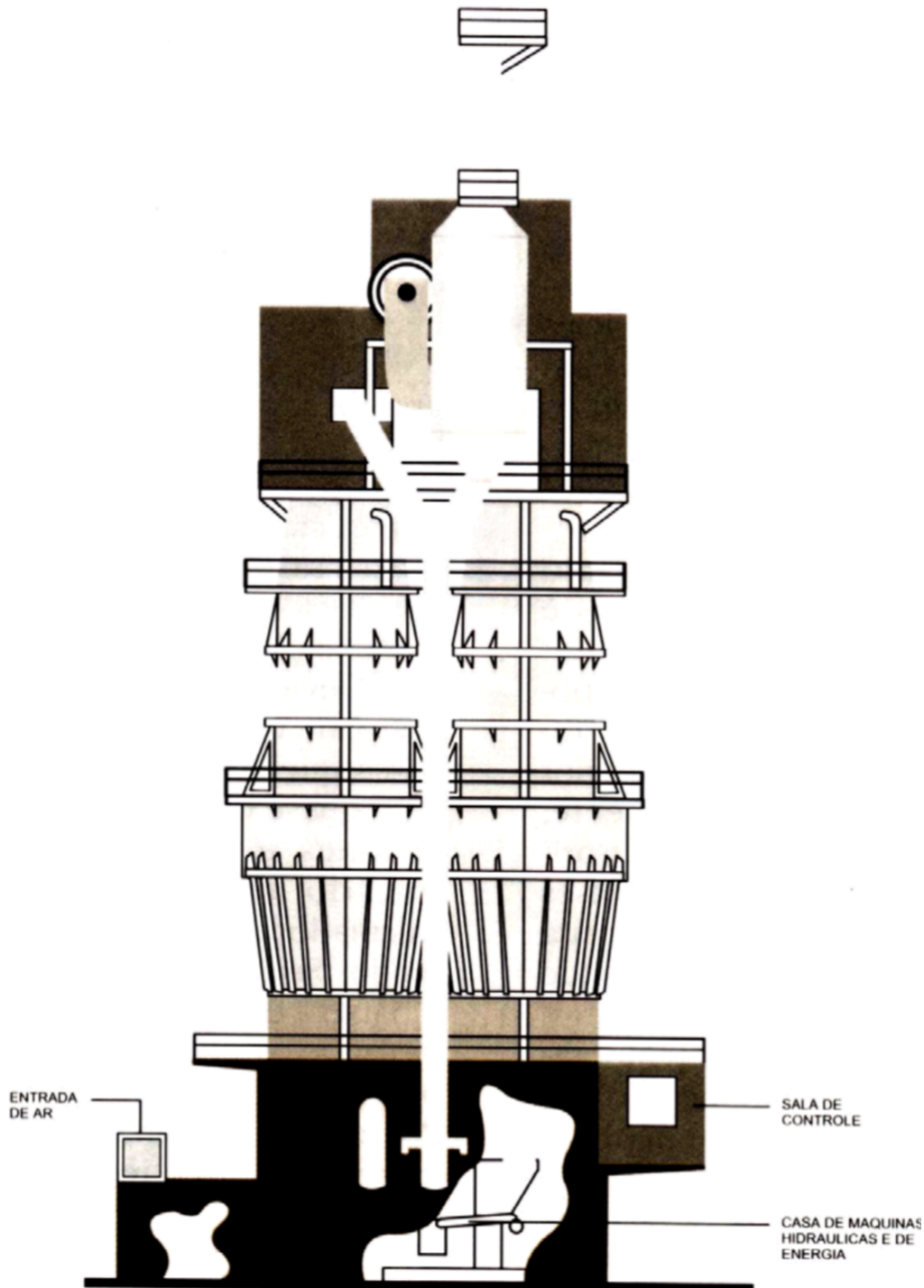


Figura 2.8 - Forno industrial de calcinação *Maerz Ofendau Parallel Shaft* (Guimarães, 2002).

A produção de finos nas operações de lavra e tratamento de calcário destinado à calcinação é um fator que onera o custo do produto devido ao seu volume excessivo. Muitos fabricantes de equipamentos procuram resolvê-lo. Por exemplo, a *Smidth* desenvolveu um processo de calcinação de finos em suspensão numa corrente gasosa (*gas suspension calcining of fines*), adequado à produção de cal, com alimentação da ordem de 1 mm para calcários calcínicos, dolomíticos e magnesianos (Guimarães, 2002).

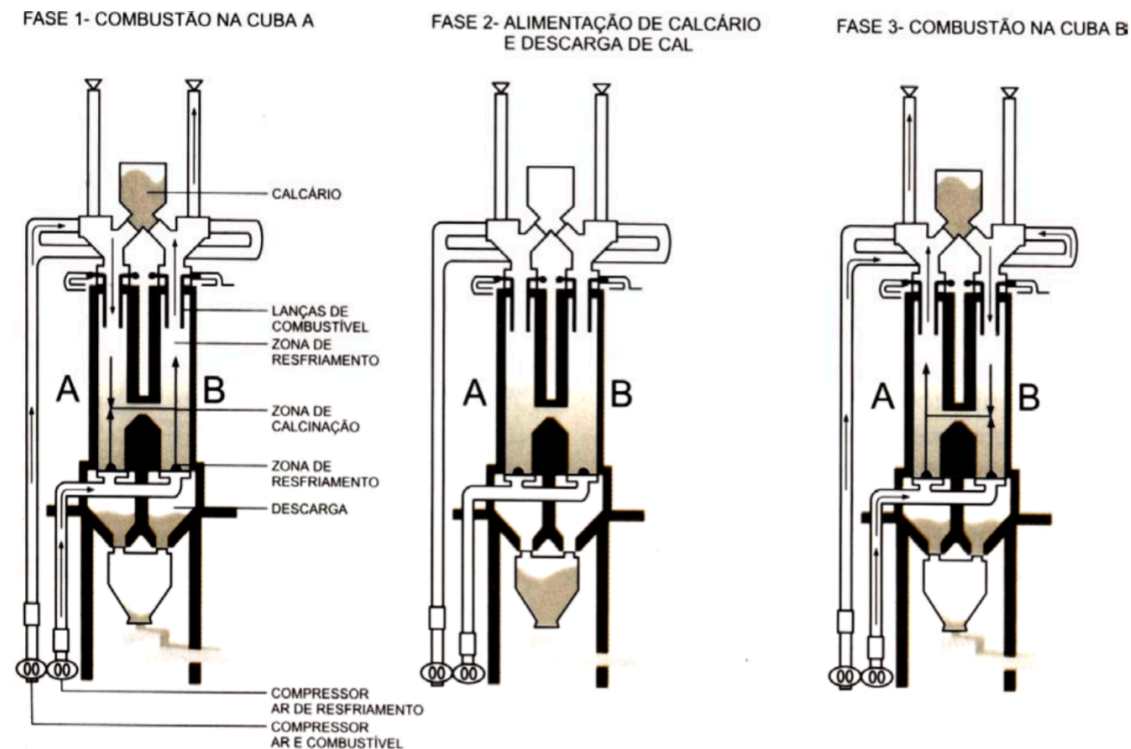


Figura 2.9 - Esquema de funcionamento de um forno de calcinação, modelo *Parallel shaft*, produzido pela *Maerz Ofendau* (Guimarães, 2002).

A carga é alimentada por meio de um transportador de copos a um conjunto de ciclones, composto por ciclones pré-aquecedores, calcinadores e recuperadores de calor. A temperatura de calcinação é da ordem de 1150 °C, aproximadamente, e o consumo de combustível 4605,5 kJ/kg (Smidth, 1975). Apresenta-se na Figura 2.10 um esquema do forno para calcinação de finos da Smidth.

O processo em leito fluidizado é conhecido desde 1949, o qual consiste em movimentar substâncias sólidas em corrente de gás, provocando uma reação com o veículo gasoso. A *Dow Oliver Corporation* foi a primeira empresa a obter da Standard Oil Development Corporation, detentora de tecnologia, permissão para construir aparelhos de leito fluidizado, sob o nome de Process Fluo-Solids (Kawasaki, 1983). As características técnicas do forno da Dow Oliver podem ser avaliadas pelos dados seguintes (Kawasaki, 1983):

- Granulometria da carga do forno: 0,3 a 25mm;
- Produção total de cal: 221 t/dia;
- Consumo de combustível: 4861 kJ/kg de cal;
- Temperatura de queima no ciclone: 980 °C.

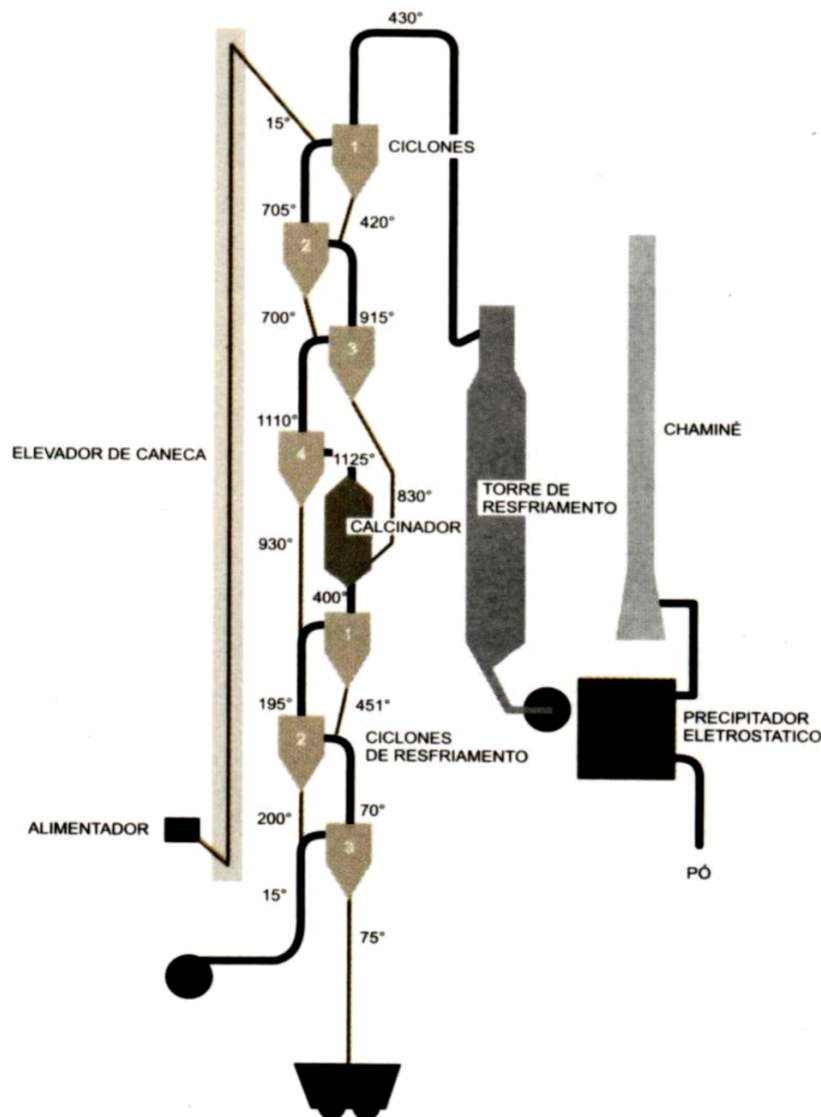


Figura 2.10 - Forno para a calcinação de finos de calcário, produzido pela *F. L. Smidth* (Smidth, 1975).

Posteriormente, outras empresas replicaram esta tecnologia, como a *Kawasaki Heavy Industries Ltda.* Um esquema do forno em leito fluidizado produzido pela *Kawasaki* pode ser visualizado na Figura 2.11 (Kawasaki, 1983).

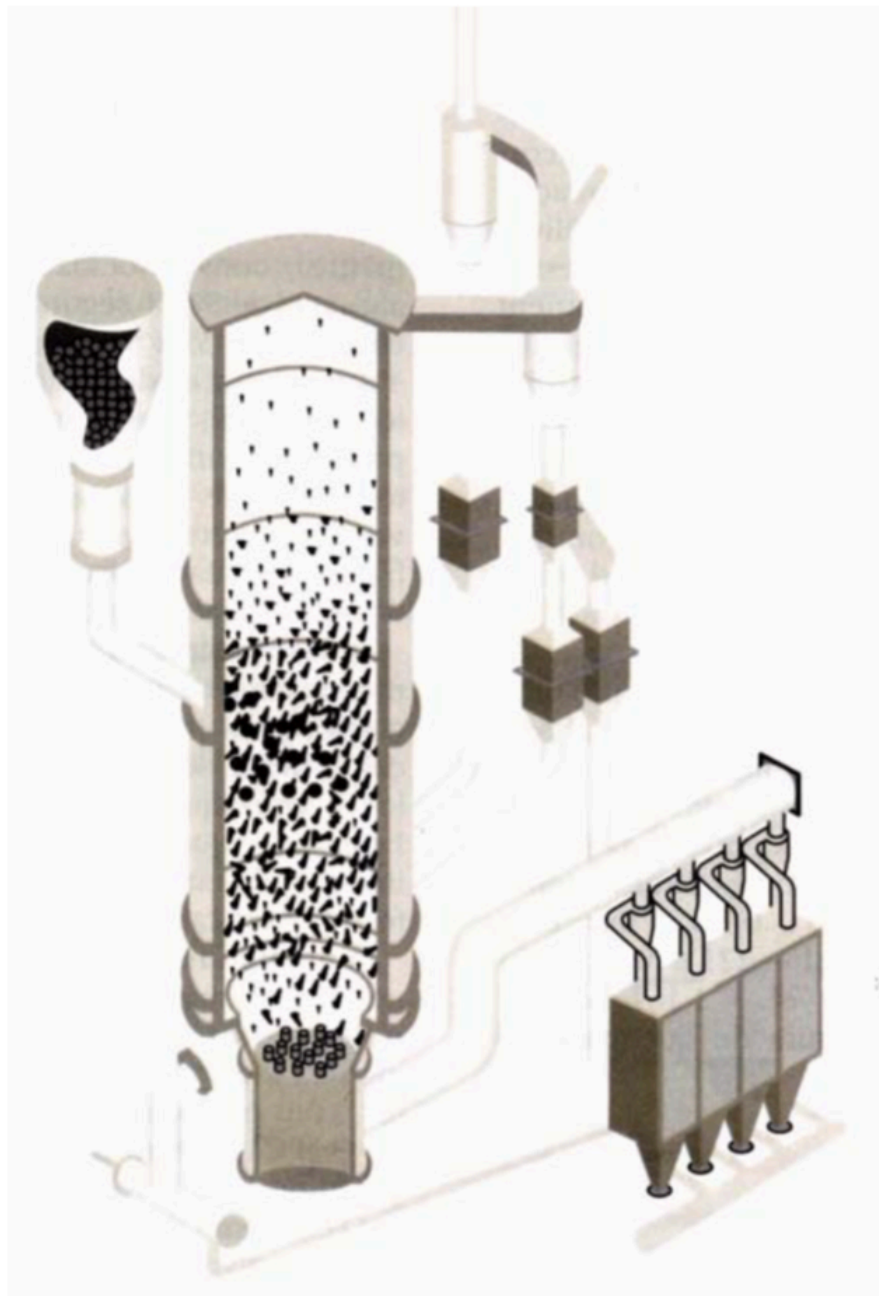


Figura 2.11 - Forno de calcinação em leito fluidizado da *Kawazaki* (Kawasaki, 1983).

Um forno de modelo patenteado pela *Calcinatic International Limited*, de Nassau, apresenta uma forma circular de grande diâmetro, câmaras de calcinação rotativas revestidas de refratários e dispostas circularmente, podendo ser operadas em várias velocidades, de 35 a 200 RPM. Após uma rotação completa, a calcinação está concluída. Através de instrumentação precisa, as temperaturas podem ser mantidas uniformes nas diferentes zonas do forno durante o giro. O controlo de temperatura e da rotação permite a calcinação de pedras de várias dimensões ou a obtenção de cal de qualquer característica desejada, isto é, de dura a macia (Kinsler, 1991). Esse modelo de forno pode ser visto através do esquema descrito na Figura 2.12.

A identificação técnica do forno, segundo o fabricante, é a seguinte:

- a. Razão granulométrica: 3:1;
- b. Temperatura de calcinação: 960 °C no interior da câmara;
- c. Consumo de combustível: 6070,9 kJ/kg de cal;
- d. Capacidade: 100, 200, 250 e 300 t/dia;
- e. CO₂ residual: menor ou igual a 1%.



Figura 2.12 - Esquema de forno horizontal de câmara rotativa desenvolvido por *Calcinatic International Limited* (Kinsler, 1991).

De origem alemã, o forno *Hoffman* ou “de anel”, foi inventado em 1865 como um modelo característico de forno horizontal de câmara fixa. O seu princípio é o de um forno túnel (circular, elíptico ou retangular), no qual os gases quentes da queima dos combustíveis sólidos percorrem lentamente o túnel, pré-aquecendo e calcinando a carga de calcário. O túnel de 2 a 3 metros de altura é dividido em 12 a 20 câmaras, carregadas manualmente através de aberturas nos arcos, formando pilares com ou sem recheio de combustível.

Dependendo da experiência do operador, a calcinação pode evoluir à razão de 0,5 m a 1,2 m por hora, passando por todas as câmaras (Schwarzkorf, 1978). Segundo Boynton (1980), este forno apresentava um menor índice de contaminação da cal pelas cinzas, numa época em que predominavam os equipamentos de cargas misturadas. A sua produção diária era da ordem de 25 a 80 toneladas, com excessiva exigência em mão-de-obra, razão pela qual o modelo teve pouca evolução e aceitação. Apresenta-se a seguir, na Figura 2.13, uma planta esquemática do forno *Hoffman*.

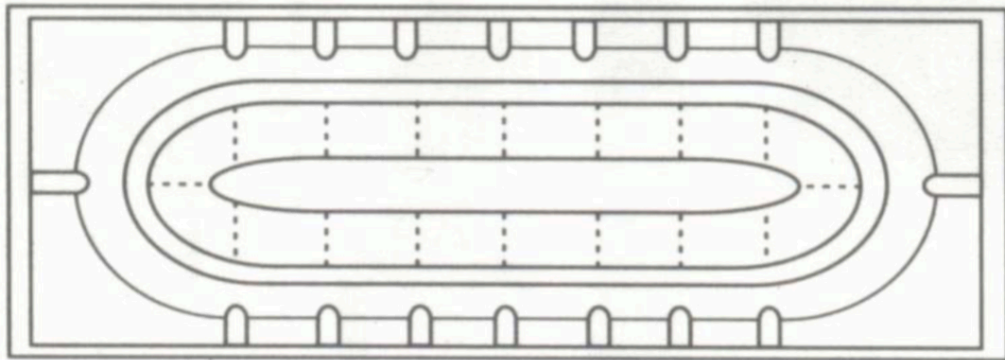


Figura 2.13 - Planta esquemática do forno *Hoffman* (Boynton, 1980).

Vários outros tipos de fornos de fluxogramas e estruturas construtivas apareceram e, por vezes, desapareceram no sector industrial de calcinação da cal. Alguns, com ligeiras modificações dos modelos tradicionais, outros com completas alterações estruturais.

Nos anos 1950 surgiu na Europa Central o forno húngaro, utilizando combustíveis de baixo poder calorífico, alto consumo energético (7451,5 kJ/t de cal) e carga de pedras de 50,8 x 76,2 mm. Na Alemanha surgiu o forno rotativo *Fellner-Ziegler*, com consumo energético de 6615 kJ/t de cal e, na Áustria, o grande forno vertical *Wopfinger Stein Kalkwerke*, de 5819,7 kJ/ t de cal de consumo de combustível (Schwarzkorf, 1978).

Schwarzkorf (1978) apresenta fornos de diferentes conceções no fluxograma da carga, na evolução da calcinação:

- i. Fornos Hereshoff: usados principalmente na recuperação de cal hidratada residual de determinadas indústrias. Neste forno a carga percorre caminhos horizontais e verticais, em câmaras (4 a 12), acionada por pás com movimentos de rotação – 0,5 a 2 RPM. É alimentado por pedras de 91,44 a 63,5 mm, que atingem 900 °C e produzem, em geral, até 500 toneladas de cal diariamente;
- ii. Fornos DSO: de dupla inclinação, capacidade de até 200 t/dia, consumo

de 4103 kJ/kg de cal, carga com granulometria de 16 a 30 mm, e temperatura de calcinação de 930 °C (câmara superior) e 1130 °C (câmara inferior). No seu interior a carga avança verticalmente passando de câmara para câmara através de uma rampa. Com uma altura de 17 m, o equipamento tem consumo energético de 48 kWh/t (172,8 kJ/kg);

- iii. Forno Kunnep-Chisaki: com o corpo em forma de anel tem movimento rotativo, intermitente e de pequeno ângulo, o que permite acumulação de carga proveniente de um silo colocado na parte superior. Vários êmbolos localizados na parede lateral empurram a carga progressivamente para o centro do corpo do forno, enquanto o movimento de rotação se processa e o material é acumulado na periferia do anel. Depois a carga é conduzida ao fundo do forno, através de zonas de aquecimento e calcinação. As suas características técnicas: produção de 25 a 30 t/dia; consumo de calor de 3977 kJ/kg de cal; combustível empregado é óleo combustível; granulometria da carga de 7 a 25 mm; rotação de 1 a 2 RPM.

As figuras abaixo apresentam esquemas desses três tipos de fornos descritos anteriormente.

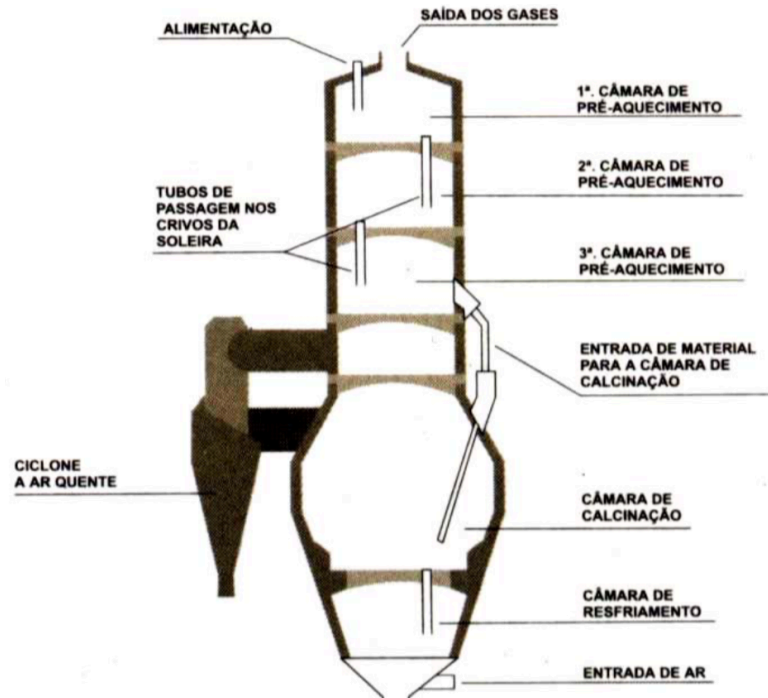


Figura 2.14 - Forno calcinador DSO (Schwarzkorf, 1978).

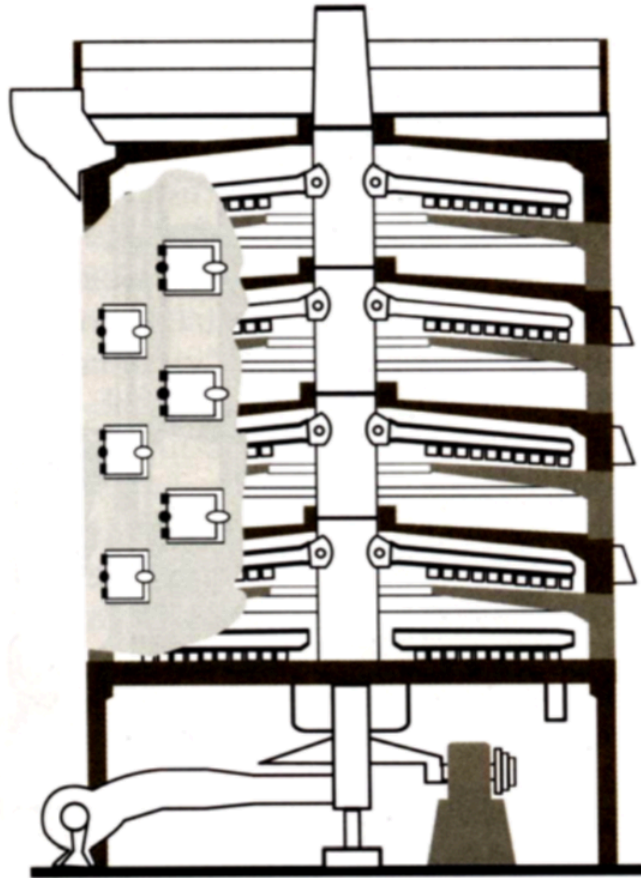


Figura 2.15 - Forno calcinador *Hereshoff* (Schwarzkorf, 1978).

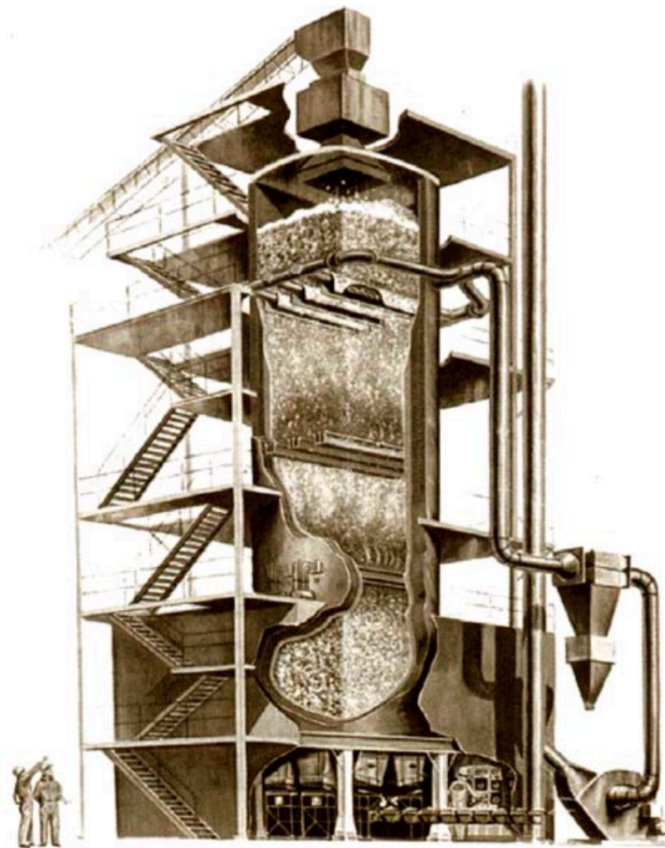


Figura 2.16 - Forno calcinador *Kunne Chisaki* (Schwarzkorf, 1978).

2.6.2.4 – Fornos rotativos

Os primeiros modelos de fornos rotativos surgiram a meados do século XIX, com capacidade de produzir 100 toneladas por dia, aceitando carga com granulometrias menores e com princípios de calcinação semelhantes aos modelos atuais. Esses modelos são apresentados de seguida.

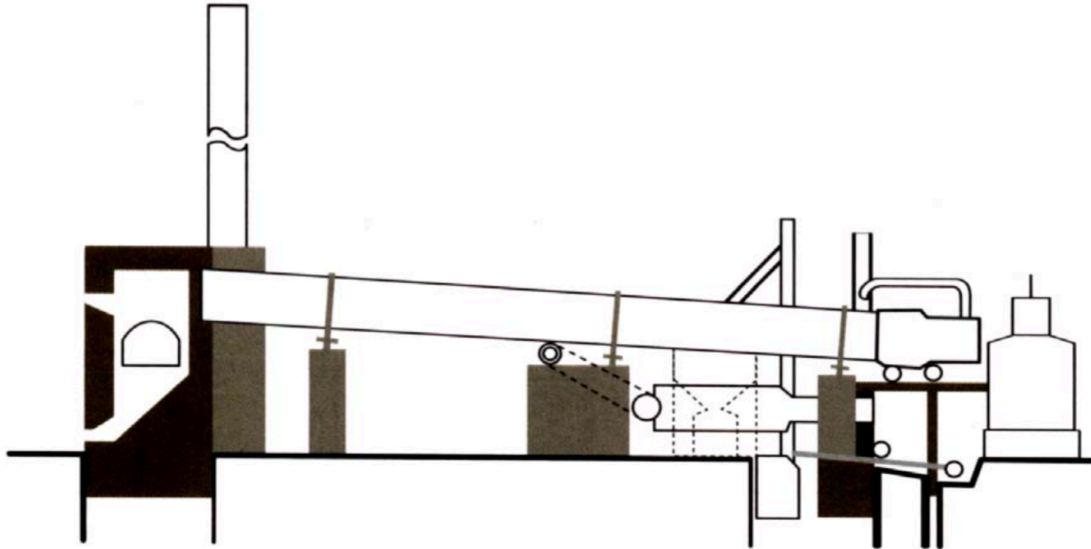


Figura 2.17 - Modelo de forno rotativo da *F. L. Smidth*, produzido na Dinamarca (Smidth, 1975).

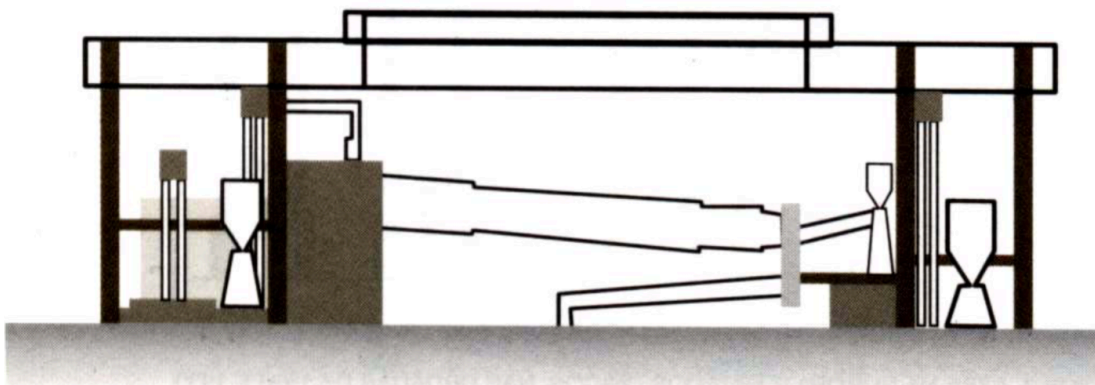


Figura 2.18 - Forno rotativo modelo *Ransome*, de 1885 (Ciaccia *et al*, 1974).

Os fornos rotativos são encontrados nos mais diversos processos que envolvem o processamento de sólidos. Estes incluem a secagem, a incineração, a calcinação, processos de sinterização, reações gás-sólidos, entre outros (Jauhari *et al*. 1998). A aplicação industrial mais comum e importante está presente na produção de cimento, sendo que a grande maioria dos produtos tem os fornos rotativos como equipamentos de escolha primordial.

Outra aplicação importante de fornos rotativos é a incineração de resíduos sólidos, são populares neste ramo devido à sua alta flexibilidade. Podem processar uma grande variedade de materiais na alimentação, com valores de

poder calorífico variável, sem apresentar maiores dificuldades na remoção dos sólidos na saída do processo. Tipicamente, os incineradores de resíduos sólidos operam com leitos relativamente profundos, e possuem uma câmara de combustão secundária após o forno rotativo para melhorar a combustão heterogênea dos resíduos (Rovaglio *et al.* 1998).

Os primeiros estudos publicados referentes a fornos rotativos estabeleciam a relação de interdependência entre a velocidade de rotação do forno, a sua inclinação axial, a profundidade do leito de sólidos e o tempo de residência do material (Sullivan *et al.*, 1927). Posteriormente foi desenvolvido um modelo baseado na suposição de que as partículas do leito desenvolviam um movimento circular, acompanhando a rotação do forno, até que essas partículas rolavam umas sob as outras para a parte inferior do leito formando uma camada fina de sólidos (Saeman *et al.*, 1951). O tempo de queda das partículas foi assumido suficientemente baixo quando comparado ao tempo para a partícula se mover da metade inferior do leito para a metade superior do leito de sólidos do forno. Usando a geometria de fornos rotativos com inclinação axial, foi observado que o ângulo de inclinação necessário para se manter uma altura constante do leito ao longo do comprimento do forno rotativo podia ser determinado em função de uma dada velocidade de rotação. Este modelo básico permitiu uma predição bem realista dos dados originais (Sullivan *et al.*, 1927). Posteriormente, o modelo foi refinado de modo a predizer o movimento axial das partículas com diferentes enchimentos de leito, levando em conta o tempo de queda das partículas para a superfície inferior do leito (Kramers *et al.*, 1952). Um trabalho posterior, que tinha por objetivo primordial a verificação do movimento das partículas na superfície interna do forno, bem como a validação do modelo de Kramers *et al.* (1952), procurou o estabelecimento de um modelo com menos restrições e maior aplicabilidade (Lebas *et al.*, 1995). Mais recentemente, o mesmo modelo fundamental para a trajetória das partículas no interior de fornos rotativos foi validado, através de um estudo do movimento das partículas em fornos rotativos sem escoamento axial (Spurling *et al.*, 2000).

Quase sempre possuem permutadores de calor transversais ao longo de seu corpo, para uma melhor transferência de calor, além de recuperadores de calor e pré-aquecedores de calcário. Estes pré-aquecedores têm um diâmetro invariavelmente menor do que o dos fornos correspondentes. O consumo de combustível (óleo ou carvão pulverizado), tomado como base um forno de 500 toneladas/dia é o seguinte: fornos longos de 6699 a 7117,6 kJ/kg e nos fornos curtos com pré-aquecedor de 5066 a 4940 kJ/kg; com melhoria devida aos trevos que os seccionam (Manfred *et al.*, 1976).

Na Figura 2.19 apresenta-se um exemplo típico de um forno rotativo de cal de comprimento longo.

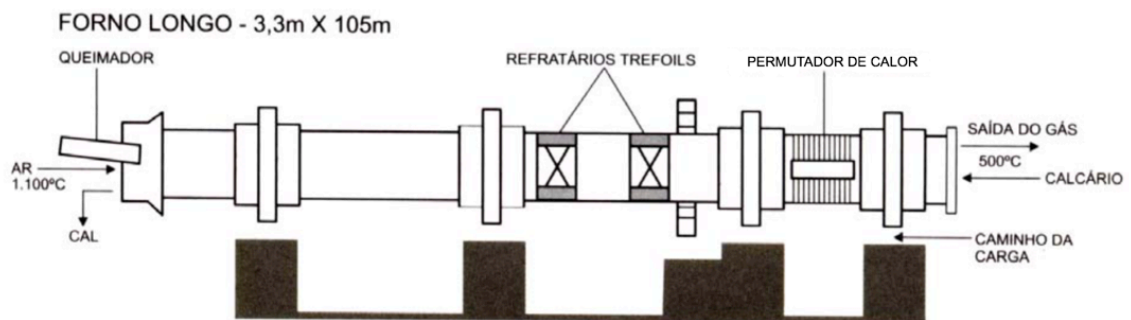


Figura 2.19 - Esquema de forno rotativo longo (Manfred *et al.* 1976).

Capítulo 3

Forno vertical

Neste capítulo é apresentada uma introdução sobre o forno vertical, desde a sua seleção até à sua geometria. Uma descrição do processo de calcinação e possíveis problemas ocorrentes durante o processo neste tipo de forno é também tido em conta neste capítulo. Por fim, é apresentado uma comparação entre o forno vertical e o forno rotativo.

3.1. – Introdução

A escolha do forno de cal é muito importante para o produtor de cal, pois o forno deve ser adequado para que seja produzida cal com a qualidade exigida. O forno deve ter um custo de capital e custos operacionais suficientemente baixos para uma produção de cal a um preço competitivo. A sua capacidade deve ser também apropriada às necessidades do mercado. Uma grande variedade de técnicas e modelos de fornos têm sido utilizados em todo o mundo ao longo dos séculos. O conceito de forno vertical foi modernizado ao longo dos anos com uma série de projetistas, as características de alguns deles encontram-se resumidos na Tabela 3.1.

Alguns modelos são mais adequados para baixas produções (abaixo de 100 t/dia), enquanto outros podem ser usados para produções mais elevadas (até 800 t/dia). O tamanho aceitável para a pedra de calcário varia de um mínimo de 30 mm para um tamanho superior de 125 mm. Certos fornos são adequados para funcionar com combustíveis gasosos, líquidos e sólidos, enquanto outros são mais restritos.

Tabela 3.1 - Sumário das características dos tipos de fornos mais comuns (G – Gasoso, L – Líquido, S- Sólido) (European Commission, 2013).

Modelo	Combustível	Produção (t/dia)	Energia térmica (GJ/t)	Tamanho da pedra (mm)
Vertical				
Alimentação mista	S	60 – 200	3,4 – 4,7	20 – 200
Coluna simples	G, S	60 – 200	3,4 – 4,7	20 – 200
Eixo Anular	G, L, S	80 – 300	3,3 – 4,9	40 – 150
Tradicional	G, L, S	100 – 600	3,1 – 6,5	25 – 200
Multi-câmaras	G, L, S	40 – 225		20 – 150
Fluxo paralelo regenerativo	G, L, S	100 – 600	3,2 – 4,2	10 – 200
Rotativos				
Longo	G, L, S	160 – 1500	6,0 – 9,2	Pó – 60
Com pré-aquecedor	G, L, S	100 - 1500	5,1 – 7,8	10 – 60

Diferentes tipos de fornos verticais têm sido desenvolvidos para reduzir o custo de energia e aumentar a produtividade:

- Forno vertical normal;
- Forno com inclinação dupla;
- Forno vertical de alimentação mista;
- Forno vertical anular;
- Forno de fluxo paralelo regenerativo.

Na indústria de calcário europeia utilizam-se principalmente fornos verticais (European Commission, 2013). Havia cerca de 211 fornos de produção de cal na União Europeia em 2003, distribuídas por todos os estados membros da UE como mostra a Tabela 3.2.

Tabela 3.2 - Número de Instalações de produção de cal na União Europeia em 2003 (European Commission, 2013).

País	Produção de Cal
Alemanha	39
Áustria	6
Bélgica	6
Bulgária	6
Dinamarca	1
Eslovénia	3
Eslováquia	6
Estónia	1
Espanha	22
França	16
Finlândia	6
Grécia	19
Hungria	3
Irlanda	1
Itália	36
Letónia	1
Lituânia	1
Portugal	4
Polónia	8
Roménia	8
Reino Unido	6
República Checa	5
Suécia	7
TOTAL	211

Em 2006, havia 597 fornos para produção de cal para comercialização (European Commission, 2013).

A indústria de cal é uma indústria consumidora intensiva de energia sendo esta responsável por 30 – 60 % dos custos totais de produção. Os fornos são alimentados com combustíveis fósseis, como combustíveis sólidos, líquidos ou gasosos, e com biomassa ou resíduos. A utilização de resíduos como combustível depende da qualidade da cal pretendida, pois aqueles podem ter um efeito sobre a qualidade. Por isso as características dos resíduos têm de ser investigadas antes de estes serem utilizados. O uso de combustíveis

fósseis tem vindo a diminuir fazendo com que o uso de combustíveis derivados de resíduos tenha crescido ao longo dos últimos anos. A Tabela 3.3 mostra a distribuição dos tipos de combustível utilizada nos fornos na UE – 27 em 2003 (European Commission, 2013).

Tabela 3.3 - Distribuição de combustível usada na queima de cal na EU-27 em 2003 (European Commission, 2013).

Tipo de combustível	EU - 27 2003 [%]
Gás (fóssil)	43
Sólido (fóssil)	41
Líquido (fóssil)	7
Resíduos	8
Biomassa	1
Outros	-

3.2. – Seleção de forno

A seleção de um forno de cal para uma instalação em particular é um processo complexo. Um ponto de partida são as características da pedra calcária. O tamanho de partículas, juntamente com o teor de calcário e as suas propriedades de calcinação podem limitar a escolha de forno. O próximo passo é estabelecer, o mais claramente possível, a qualidade da cal necessária, em termos do teor de CaCO_3 , teor de enxofre, reatividade e outros parâmetros-chave. A disponibilidade e custo dos combustíveis adequados é um fator importante, o qual interage com a escolha do forno (Oates, 1998). Outros fatores, tais como restrições de planeamento sobre a altura e a área disponível para a instalação de forno (e para quaisquer futuras extensões) também podem limitar a escolha de forno.

Depois disso, as considerações de projeto padrão (por exemplo, os custos de capital/operacionais e logística de número/tamanho de fornos em relação à procura atual e previsão do mercado) são igualmente fatores importantes a levar em conta. A medida em que um novo forno pode encaixar-se na infraestrutura existente e, particularmente, quanto ao manuseamento dos equipamentos de cal e calcário e armazenamento existentes, também pode ter um efeito significativo sobre o custo de capital e facilidade de instalação (Oates, 1998).

Muitos produtores de cal optam por instalar os dois tipos de fornos, os verticais (para granulometrias maiores de calcário) e os fornos rotativos (para os tamanhos menores). Isso faz um bom uso da pedra processada e produz uma gama de tipos e qualidades que pode atender uma variedade de procura do mercado. Uma tendência recente, no entanto, tem sido a de se instalar fornos verticais concebidos para aceitarem tamanhos menores de pedra (Oates, 1998).

3.3. – Forno vertical normal

O forno vertical tradicional é o mais básico e mais antigo dos fornos verticais, tem sido usado em muitas indústrias durante os séculos. Ele opera continuamente e é alimentado com combustível introduzido na zona de calcinação. Um diagrama esquemático que representa um forno vertical tradicional é dado na Figura 3.1. Os combustíveis utilizados têm sido vários, como óleos combustíveis, gás natural e carvão betuminoso. Ao longo dos últimos anos este tipo de forno sofreu várias melhorias na sua operação, estes desenvolvimentos foram feitos principalmente por tentativa e erro.

No forno vertical o combustível é injetado na parede do forno, geralmente não penetra mais do que 1 metro para dentro da carga. Isso limita o diâmetro interior do forno e restringe a capacidade de produção.

Foram usadas várias técnicas para permitir um aumento do diâmetro do forno e conseqüentemente um aumento na capacidade de produção. Em fornos alimentados com gás manufaturado, o volume de gás de baixo poder calorífico favorece a penetração, a qual é frequentemente assistida por injeção de ar adicional na zona de calcinação. Alguns fornos de queima de óleo usam gases de escape para aumentar a penetração do óleo vaporizado. Noutras configurações em que se queima gás, utiliza-se apenas 50% do ar estequiométrico na câmara de combustão, que é injetado a 1 m para dentro da carga através de tubos arrefecidos por água permitindo deste modo que o forno tenha 4 m de diâmetro (Oates, 1998).

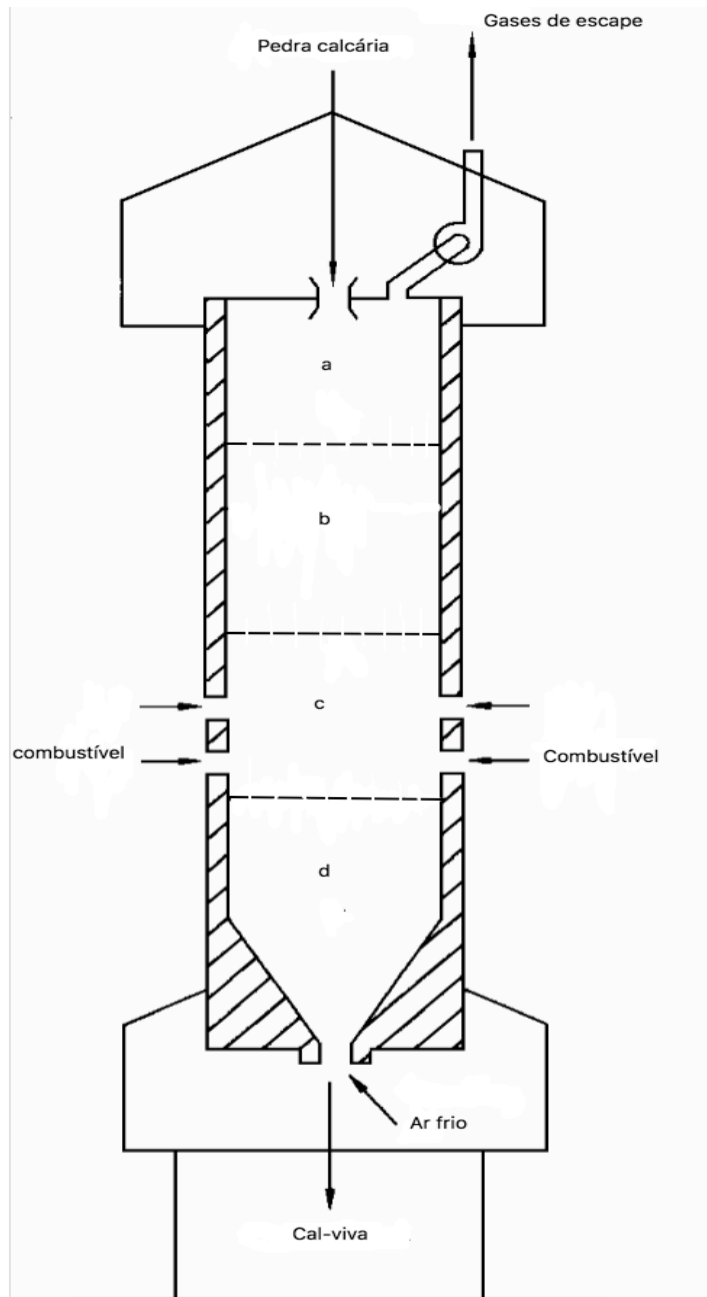


Figura 3.1 - Seção transversal de um forno vertical normal (Oates, 1998).

3.4. – Características geométricas

O forno vertical tradicional tem alturas, diâmetros e detalhes de construção variando de acordo com o tipo de aplicação, mas o forno com formato de um cilindro é o tipo mais comum. Em geral, a razão do diâmetro e a altura pode variar entre 1:2,5 a 1:5, mas o seu dimensionamento depende da sua aplicação. Tipicamente, a altura varia de 3 a 22 metros e o diâmetro interior varia de 1,5 a 4 metros. O interior é revestido com duas camadas de tijolo refratário e isolante por várias razões, mas o principal efeito é suportar a

temperatura de funcionamento e para economizar energia. A descarga da cal é feita por portas de saída com cerca de 60 cm de largura e 75 cm de altura.

3.5. – Descrição do processo

O forno vertical tradicional trabalha sob um princípio muito simples, é carregado pelo topo com pedra de calcário, a queima dá-se na zona de calcinação, enquanto o ar é injetado a partir da parte inferior do forno. A carga sólida desloca-se em sentido descendente por gravidade, passando através de quatro zonas:

a. Armazenamento: esta zona está localizada no topo do forno e age como um funil ou silo de armazenamento por onde recebe a pedra de calcário;

b. Pré-aquecimento: esta zona utiliza os gases quentes provenientes da zona de calcinação, aquecendo-se assim o calcário. Deste modo recupera-se o calor existente nos gases de combustão e calcinação. Estes gases ao entrarem em contacto com o calcário, que apresenta um movimento descendente, vão aquecendo as pedras até que estas entrem na zona de calcinação. Nesta zona a pedra é aquecida até à temperatura de dissociação, atingindo temperaturas na ordem dos 600 °C a 800 °C;

c. Calcinação: Nesta zona a temperatura encontra-se na gama de 900 °C – 1200 °C, nesta gama de temperaturas ocorrem vários processos complexos – a queima do combustível faz com que a pedra calcária se decompõe em cal (CaO) e dióxido de carbono (CO₂). Na zona de calcinação pelo menos 95% da cal é queimada, isto é, calcinada. A parte inferior desta zona é por vezes chamada de zona de acabamento, pois é onde a calcinação se completa. A fase de calcinação é responsável por 50% dos custos de produção (Ochoa, 2010);

d. Arrefecimento: O ar frio entra nesta zona a partir da base do forno, ou ponto de descarga, e por sucção passa em contracorrente com a cal que vai descendo do forno. O ar arrefece a cal recuperando-se assim a energia térmica disponível agora nas pedras de cal e entra posteriormente, como ar secundário, para a zona de calcinação imediatamente acima da zona de arrefecimento. A cal é assim arrefecida gradualmente até ser descarregada. (Figura 3.1).

Na Figura 3.2 mostram-se as quatro zonas térmicas existentes no forno vertical assim como os perfis de temperatura da superfície da pedra e dos gases. O dióxido de carbono na fase gasosa é produzido pela combustão do combustível e pela decomposição de calcário.

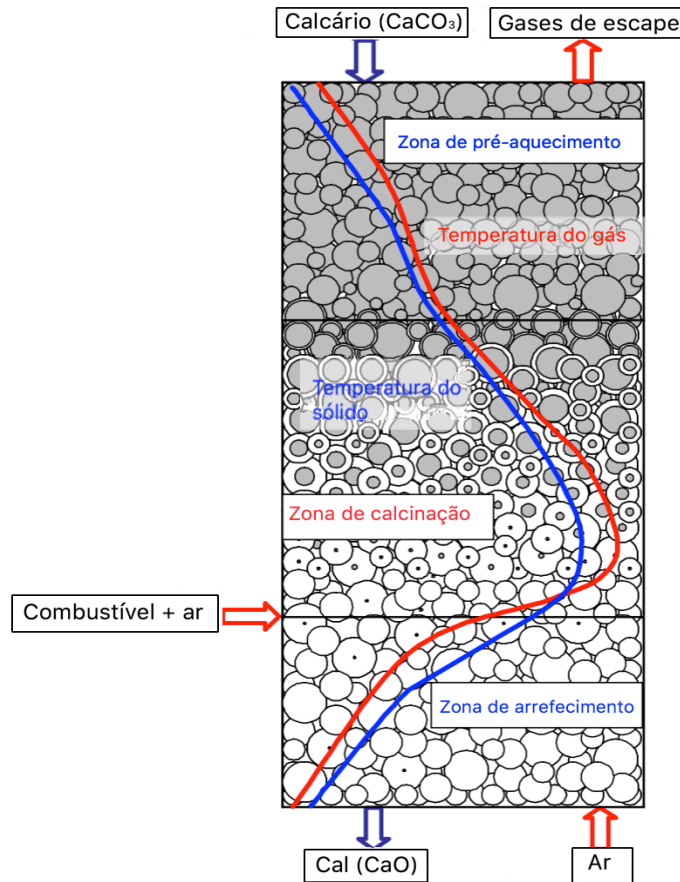


Figura 3.2 - Perfil de temperatura num forno vertical.

Em suma, a zona de pré-aquecimento utiliza os gases de escape (quentes) provenientes da zona quente, enquanto a zona de arrefecimento pré-aquece o ar de combustão maximizando assim a recuperação de calor e aumentando a eficiência térmica do processo.

3.6. – Problemas ocorrentes

Embora, os fornos verticais de cal tenham evoluído ao longo dos séculos tem havido pouca investigação sobre o seu funcionamento e os fatores que afetam o seu desempenho (Oates, 1998). Para o bom funcionamento de um forno, é necessário um fornecimento de combustível e calcário adequado. Os principais problemas que podem ocorrer em operações de fornos verticais são discutidos de seguida.

3.6.1. – Baixa eficiência do forno

As causas habituais da baixa eficiência do forno são as seguintes:

a. Baixa qualidade do calcário: O calcário é um recurso natural, a estrutura e composição varia de camada para camada dentro da mesma pedreira. Consequentemente, isso faz com que a qualidade do calcário varie ao longo do tempo de exploração da pedreira. O calcário consiste principalmente de carbonato de cálcio CaCO_3 e algumas impurezas (MgCO_3 , SiO_2 , Al_2O_3 , Fe_2O_3 , Na_2O , SO_4 , etc.). Calcário de baixa qualidade refere-se ao seu baixo teor de CaCO_3 e altas impurezas. É geralmente difícil estimar as distribuições espaciais de impurezas de calcário deste modo não se pode evitar ou remover (Koike *et al.*, 2003). Portanto, a temperatura na zona de calcinação do forno deve ser mantida a cerca de 1200 °C. A alta temperatura, acima de 1300 °C, as impurezas presentes no calcário formam clínquer (material duro/tipo de cimento) que pode bloquear o forno e também danificar o revestimento de tijolo refratário. Na Figura 3.3 mostra-se uma fotografia de uma escória cristalina retirada de um forno de cal.

b. Tamanho do calcário não uniforme: Os tamanhos da pedra calcária desempenham um papel importante na calcinação. As grandes partículas são difíceis de calcinar uniformemente e requerem um maior tempo de processamento. Se o tempo for curto, o calcário não é completamente calcinado, isto é, a eficiência do forno é reduzida. Em contrapartida, pequenas partículas aumentam a resistência ao escoamento da fase gasosa, deste modo a qualidade de preparação do calcário é um dos fatores que determina a eficiência da operação do forno. Um dos métodos mais eficientes para a produção de cal de alta qualidade é a remoção de finos da matéria-prima carregada no forno (Gamej *et al.*, 2004).

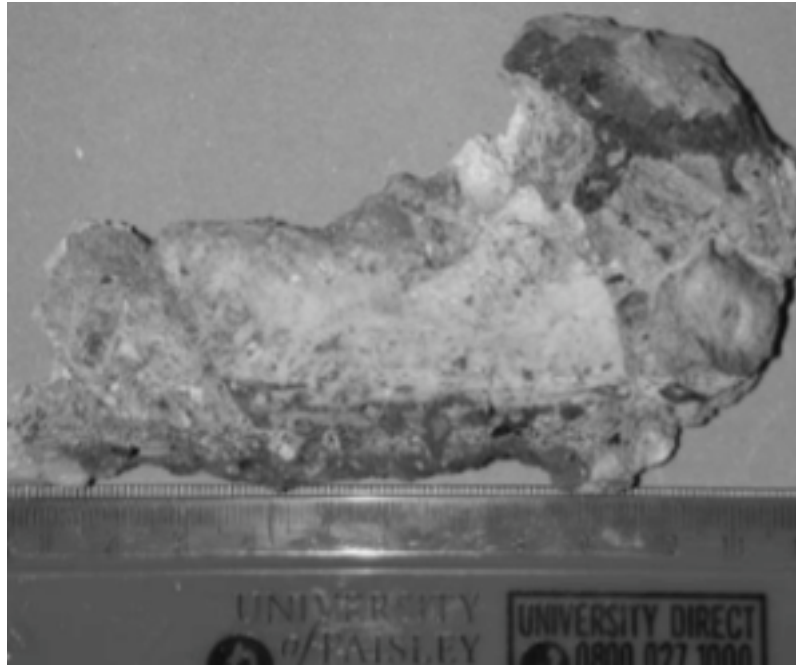


Figura 3.3 - Fotografia de um fragmento de uma escória cristalina com 70 mm de diâmetro (Leslie et al., 2004).

3.6.2. – Calcinação incompleta

O calcário não processado (calcinado) deve-se à imperfeição das reações de conversão no interior do forno, uma grande quantidade de não queimado reduz a qualidade da cal. Os parâmetros que fazem com que haja calcinação incompleta são:

- a. Ar insuficiente causando uma combustão incompleta, e consequentemente, a temperatura no interior do forno é inferior à desejada, não calcinado o calcário;
- b. O tempo de residência de grandes partículas de calcário não é suficiente para que fiquem totalmente calcinadas e, consequentemente, apresentam maiores resíduos de carbonatos;
- c. A mistura incompleta do calcário com a corrente de gases de combustão causa uma calcinação insuficiente do calcário em algumas partes do forno e um excesso de queima da cal noutras partes do forno. A Figura 3.4 mostra a seção transversal de partículas de calcário incompletamente calcinadas extraídas de um forno. É visível na figura que as partículas têm um núcleo cinzento-escuro que não reagiu, rodeado por um invólucro calcinado branco.

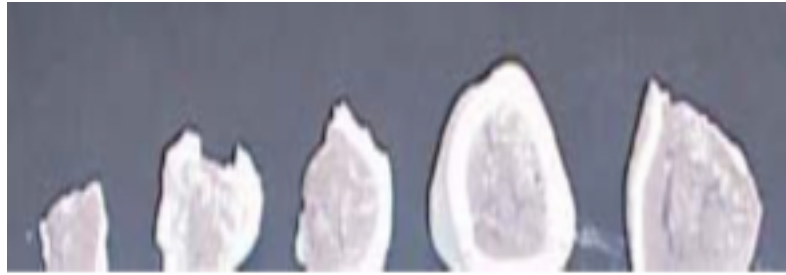


Figura 3.4 - Seção transversal de partículas de cal não calcinadas (Tran *et al.*, 2005).

3.6.3. – A influência do tipo e tamanho do combustível no desempenho do forno

Tradicionalmente, o coque tem sido o combustível mais usado na indústria de cal. As propriedades do coque variam com as propriedades do carvão e a própria concepção do forno, bem como as condições de carbonização usadas. Logo, a qualidade do coque influencia a operação do forno.

Com tamanhos de partículas de coque não uniformes, a altura das zonas do forno mudam consoante as taxas de queima das partículas de combustível. Se as partículas de coque forem pequenas estas queimam rapidamente, passando-se o oposto para partículas de maiores tamanhos. Partículas de combustível demasiado pequena provocam alterações nas zonas de pré-aquecimento e calcinação e os gases de escape com altas temperaturas e consequentemente energia não são devidamente aproveitados. Partículas grandes continuam a queimar mesmo depois de passarem a zona de calcinação, causando a mudança das zonas de calcinação e arrefecimento. Consequentemente, a cal descarregada do forno ainda se encontra demasiado quente.

3.6.4 – Alto teor de CO nos gases de escape

No caso de utilização de coque, este reage com o dióxido de carbono (CO_2) para formar monóxido de carbono (CO). As altas temperaturas e não havendo oxigénio suficiente para a combustão dar-se-á a reação $\text{C} + \text{CO}_2 \rightarrow 2\text{CO}$. Por isso o coque deverá ter uma baixa reatividade em relação à redução de dióxido de carbono, o que é muito difícil de se conseguir na prática. Consequentemente o controlo do excesso de ar é fundamental na operação dos fornos de calcinação para minimizar fenómenos de combustão incompleta ou redução do CO_2 .

3.7. – Combustível

Fornos verticais tradicionais operam continuamente e são alimentados com combustível introduzido na zona de calcinação. Vários combustíveis têm sido utilizados, inclusive carvão betuminoso, gás manufacturado, óleo combustível e gás natural.

O carvão é introduzido no forno na parte superior da zona de calcinação, quer por operadores, por pá mecânica ou por projeção, num espaço criado por um estreitamento da haste. As camadas de carvão e calcário são alternadamente introduzidas, por vezes o carvão é misturado com o calcário antes de serem introduzidos no interior do forno. O carvão tem que ter uma temperatura de fusão de cinzas adequada, de preferência acima de 1200 °C, o que se consegue através de um baixo teor de cinzas aceitável. A alta temperatura de fusão de cinzas reduz o risco de sinterização das partículas de cal, causada pelas cinzas de fusão e pela interação entre os pedaços de cal e os refratários. É também assegurado que a maior parte das cinzas não aderem à cal e possam ser facilmente removidas (Oates, 1998). O carvão utilizado tem de possuir propriedades de coqueificação adequadas de modo que se formem aglomerados coesos, que queimem de forma relativamente lenta e fiquem presos entre os pedaços de calcário em calcinação.

A maior parte da matéria volátil do carvão é destilada logo após a queima e queimada na zona de pré-aquecimento, ou é emitida para a atmosfera como gases de escape. Por outro lado, a maior parte do enxofre no carvão é absorvido pela superfície da cal, formando sulfato de cálcio. Devido à sua maior área de superfície por unidade de massa, a concentração de enxofre é maior nas partículas de cal com menores dimensões (Oates, 1998).

Gás manufacturado também é usado para a produção de cal. É obtido pela combustão parcial do carvão (ou óleo) com defeito de ar. Vapor de água também pode ser adicionado a este processo de produção de gás combustível. Os principais componentes combustíveis do gás manufacturado são o monóxido de carbono e o hidrogénio, podendo ainda apresentar com uma quantidade mínima de metano. Contudo, os gases inertes perfazem a maior parte do volume do gás. Eles consistem em azoto com pequenas quantidades de dióxido de carbono, vapor de água e oxigénio. O enxofre no combustível é convertido em sulfureto de hidrogénio ou em dióxido de enxofre. A maior parte da cinza do carvão original fica no sistema de produção do gás manufacturado, de forma que a cal não fica contaminada (Oates, 1998). Com um baixo poder

calorífico volúmico do gás seco (cerca de 6280 kJ/m^3 a 0°C e 1 atm), o gás manufaturado é um combustível conveniente para a queima de cal porque as suas chamas longas e lentas produzem uma cal muito reativa (Oates, 1998).

O óleo combustível é, geralmente, gasificado em câmaras refratárias alinhadas e ligadas ao forno, ou vaporizado por pulverização sobre a superfície das partículas de calcário ou cal. É geralmente queimado com um baixo nível de excesso de ar primário para produzir chamas lentas e relativamente longas, necessárias para uma calcinação controlada. No entanto, a cal produzida tende a ser de moderada a baixa reatividade. Em fornos verticais, a maior parte do enxofre no óleo é retida como sulfato de cálcio na superfície dos pedaços de cal. O óleo combustível é isento de cinza. A operação com um excesso limitado de ar resulta na emissão de alguns voláteis e negro de fumo. Isto, juntamente com a relativa baixa reatividade do calcário tem causado muitos fornos verticais movidos a óleo serem substituídos por fornos mais modernos.

O gás natural é introduzido lateralmente no forno. Queima mais tempo, com uma chama mais lenta do que o petróleo ou o carvão e produz uma cal de reatividade média. Como contém quantidades insignificantes de enxofre e não produz cinzas, a cal resultante não é contaminada pelo combustível em qualquer momento. Além disso, o gás natural não produz fumo, de modo que os gases de escape têm uma baixa opacidade. O calor empregue em fornos verticais a gás natural é, geralmente, de 4815 kJ/kg (Oates, 1998).

Quando o combustível é injetado na parede de um forno, geralmente não penetra mais do que 1 m para dentro da carga. Isso limita a largura eficaz do forno, ou o diâmetro, a cerca de 2 m , e restringe a capacidade produtiva de um eixo de diâmetro 2 m a cerca de 80 t/dia (Oates, 1998).

Ao longo dos anos foram usadas várias técnicas para permitir que o diâmetro do forno, e portanto a sua capacidade de produção fosse aumentada. Em fornos consumindo gás manufaturado, o grande volume de gás de baixo valor calorífico favorece uma maior penetração e é frequentemente assistida pela injeção de ar adicional, ou "secundário" para a zona de calcinação.

Em muitos países, o aumento generalizado na disponibilidade de gás natural a preços competitivos, nos anos 60 e 70, permitiu que muitos produtores de cal, para estender a vida operacional de seus fornos (verticais tradicionais), optassem pelo gás natural como combustível de operação. O gás natural traz três grandes vantagens:

- a. É livre de enxofre, de modo que a cal atende aos requisitos de enxofre de siderúrgicas;
- b. Queima com uma chama longa e lenta, que tem uma emissividade relativamente baixo. Isso permite a produção de uma cal mais reativa para uma determinada percentagem de CaCO_3 , ou permite que a percentagem de CaCO_3 seja reduzida para uma dada reatividade;
- c. Não produz fumos quando queimado em condições normais de operação (Oates, 1998).

As vantagens acima referidas, juntamente com o desenvolvimento de técnicas para melhorar a penetração da queima de gás no centro do forno, permitem fornos verticais a gás natural com diâmetros de até 5 m e produções de cal de mais de 200 t/dia (Oates, 1998).

3.7.1. – Combustão

Em todos os processos de combustão há uma razão ar – combustível ótima que maximiza o rendimento da combustão. Uma razão mais baixa resulta numa combustão incompleta e um aumento dos níveis de monóxido de carbono, enquanto uma razão elevada faz com que os produtos de combustão são diluídos e arrefecidos pelas quantidades adicionais de ar. Por isso na pratica há sempre que se adotar uma solução de compromisso.

A combustão dentro de um leito fixo, como é o caso dos fornos de cal verticais é particularmente problemática, pois a mistura entre o combustível e ar nestas condições é um processo ineficiente. Do ponto de vista da eficiência de combustão, o combustível e o ar devem, idealmente, ser distribuídos uniformemente em todo leito. No entanto, independentemente do sistema de combustão utilizado, as variações na razão de ar-combustível podem ocorrer. Do ponto de vista da calcinação do calcário, as chamas, lentas e longas resultantes da combustão com uma deficiência de ar são mais favoráveis do que, chamas intensas e curtas, isto é, porque uma deficiência de ar reduz as temperaturas na zona de calcinação, particularmente na secção de "acabamento" e ajuda a garantir uma elevada reatividade. Mas a deficiência de ar leva à combustão incompleta com formação de monóxido de carbono e

outros não queimados reduzindo-se assim o rendimento do processo (Oates, 1998).

3.8. – O forno vertical comparativamente ao forno rotativo

Uma das principais características operacionais dos fornos rotativos é a maneira peculiar como a carga deste se movimenta no seu interior, a rotação imposta ao forno faz com que a carga sólida acompanhe o movimento das paredes até certa altura e então caindo e rolando de forma caótica sobre as paredes. A rotação é impressa sobre o forno por meio de motores elétricos externos, geralmente de rotação variável e o mecanismo de transmissão de potência pode contar com correias ou cremalheiras. O custo operacional dos fornos rotativos é mais elevado (comparado com os fornos verticais), além do custo do consumo dos motores externos que permitem a rotação do forno, há o acréscimo de custo devido a manutenção de todo o sistema mecânico.

Devido ao movimento da carga no interior do forno rotativo, pode ocorrer a suspensão de partículas na atmosfera, no interior do forno. Isto é, se a granulometria da carga for muito fina, ou se esta se esfarelar com muita facilidade. É assim possível haver a formação de nuvens de partículas, o que é extremamente indesejado, pois podem ser carregadas com o escoamento de gases para fora do forno.

A medição da temperatura no interior do forno rotativo é uma tarefa bastante difícil, devido às suas características construtivas e ao contínuo movimento, que inviabiliza a instalação de termopares ao longo do seu comprimento. Assim, os fornos rotativos encontrados em processos industriais contam com termopares instalados apenas nas seções de entrada e saída de carga, o que implica num controle retardado do processo de queima, podendo trazer problemas, tornando incapaz de antecipar variações nas curvas de temperatura durante o processo. Por outro lado, as características funcionais do forno vertical permitem a instalação de termopares em cada zona do processo (pré-aquecimento, calcinação e arrefecimento), se necessário.

A fração do volume do forno rotativo ocupado pela carga tem uma forte influência sobre os processos de transferência de calor e de massa, e sobre o tempo de residência da carga no interior do forno, sendo os fornos rotativos

operados, em geral, com baixos níveis de preenchimento comparativamente aos fornos de eixo vertical. O tempo de residência é muito mais curto num forno rotativo do que num forno vertical. Isto traduz-se para uma taxa significativamente mais elevada de pré-aquecimento do calcário num forno rotativo.

Os fornos verticais tradicionais possuem taxas de produção relativamente baixas. Pois o combustível injetado numa parede, geralmente, não penetra mais do que 1 metro para o leito compactado, e isso limita a produtividade.

O grande problema com o forno vertical tradicional é a obtenção de libertação de calor uniforme e o movimento da carga em todo o eixo. Outro inconveniente é o facto de que o carvão não pode ser utilizado sem degradar a qualidade da cal produzida.

Capítulo 4

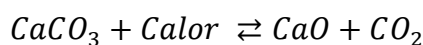
Calcinação

Neste capítulo são destacados os aspectos referentes à cinética de calcinação e fatores que a influenciam. As propriedades físicas, distribuição do tamanho e formato da partícula a calcinar são tidas em conta neste capítulo.

4.1. –Introdução

A calcinação de calcário ocorre numa zona muito estreita, quando o carbonato de cálcio é dissociado em óxido de cálcio e dióxido de carbono. Essa zona avança do exterior para o interior da pedra com uma velocidade constante para cada temperatura, independentemente do tamanho da pedra ou grau de calcinação. A maior parte da resistência à transferência de calor para a pedra parece estar na zona de calcinação e não no corpo do material calcinado (Boynton, 1980).

A decomposição térmica dos calcários resulta na formação de dois produtos – cal e dióxido de carbono. Esta reação, quimicamente bem estabelecida em relação às massas moleculares é reversível para o carbonato de cálcio. A termodinâmica fornece as duas informações básicas da calcinação, relacionadas com a reação e ligadas às condições de temperatura e pressão. Quando o carbonato de cálcio é aquecido (queimado) num forno a temperatura de 900 – 1200 °C, é decomposto em cal (óxido de cálcio e dióxido de carbono).



A reação é prontamente reversível e a “recarbonatação” da cal ocorre na presença de CO_2 . A pressão de CO_2 em equilíbrio com CaCO_3 e CaO aumenta com o aumento da temperatura. As pressões de equilíbrio são dadas na Tabela 4.1.

Tabela 4.1 - Pressão de equilíbrio do CO_2 na calcinação (Beach, 1983).

Temperatura (°C)	Pressão (atm)
700	$2,92 \times 10^{-2}$
897	0,220
902	1,0
1000	3,87

Durante a calcinação num forno aberto, o CO_2 é arrastado por correntes de convecção. Portanto, a sua pressão parcial é sempre menor do que o valor de equilíbrio e essencialmente todo o CaCO_3 se decompõe. A temperatura de dissociação do calcário é 902 °C, temperatura à qual a pressão de CO_2 é igual a 1 atm.

A pureza da cal depende da composição do calcário a partir do qual é derivada. O calcário não é 100 % calcite, uma vez que contém sempre algum carbonato de magnésio que se decompõe a cerca de 450 °C. Assim, a maioria da cal contém alguma percentagem de MgO . Cales com altos teores em cálcio são aquelas com 95 % ou mais de CaO .

4.2. – Degradação térmica

A degradação térmica pode ocorrer por três razões:

1. Algumas pedras calcárias decrepitam em temperaturas muito abaixo de 900 °C;
2. Algumas pedras calcárias são propensas a quebrar, se forem expostas a ciclos térmicos (por exemplo, como um fluxo paralelo de um forno regenerativo);
3. Outras degradam-se na calcinação.

Tal degradação reduz a porosidade do leito e pode afetar adversamente a calcinação e o arrefecimento da cal. Além disso, a quantidade de finos (tamanhos iguais ou inferiores a 5 mm) aumenta, reduzindo o rendimento dos produtos rastreados (Oates, 1998). As causas de decrepitação estão relacionadas com os efeitos do calor sobre os cristais. Calcários com grandes

cristais são muitas vezes mais propensos à crepitação do que os depósitos microcristalinos. A única abordagem satisfatória é comparar o seu desempenho com amostras que são conhecidos como sendo adequados para uma determinada conceção do forno (Ruckensteiner *et al.*, 1995).

4.3. – Distribuição do tamanho da partícula

A distribuição do tamanho da partícula deve ser compatível com os requisitos do forno. Isto requer, geralmente, que a partícula a ser positivamente rastreada tenha uma razão de tamanho de, preferencial de 2:1 ou, pelo menos 3:1. Embora alguns fornos possam tolerar uma gama ampla de tamanhos, a consequência é certamente uma redução de qualidade de produto, de saída ou maior consumo de energia térmica por tonelada de produto, causada pela (a) gama de tempo de calcinação e (b) reduzida porosidade no leito (Oates, 1998).

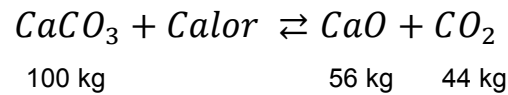
O limite superior dos tamanhos das partículas na alimentação dos fornos verticais está geralmente na gama de 75 a 150 mm, embora alguns fornos tenham sido concebidos para aceitarem tamanhos de 30 mm para baixo (Mereu, 1994).

4.4. – Formato da partícula

A pedra calcária para a queima pode ser ligeiramente achatada. Isto permite utilizar rolos giratórios e britadeiras de maxilas, que produzem uma proporção menor de produto subdimensionado e consomem menos energia por tonelada de pedra. Uma pedra achatada implica ligeiramente uma maior resistência ao escoamento de gás do que um tamanho semelhante a uma pedra mais cúbica. Isto torna necessário o uso de ventiladores capazes de suportarem maiores perdas de carga.

4.5. – Reação química

A reação para a decomposição térmica do carbonato de cálcio pode ser escrita como:



A relação entre a pressão de dissociação da calcite com a temperatura é mostrada na Figura 4.1. Esta pressão atinge 1atm (101,3 kPa) a cerca de 900 °C. Durante muitos anos, o valor de 898 °C era considerado como o valor correto, mas trabalhos de pesquisa mais recentes apontam para um valor de 902 °C (Oates, 1998).

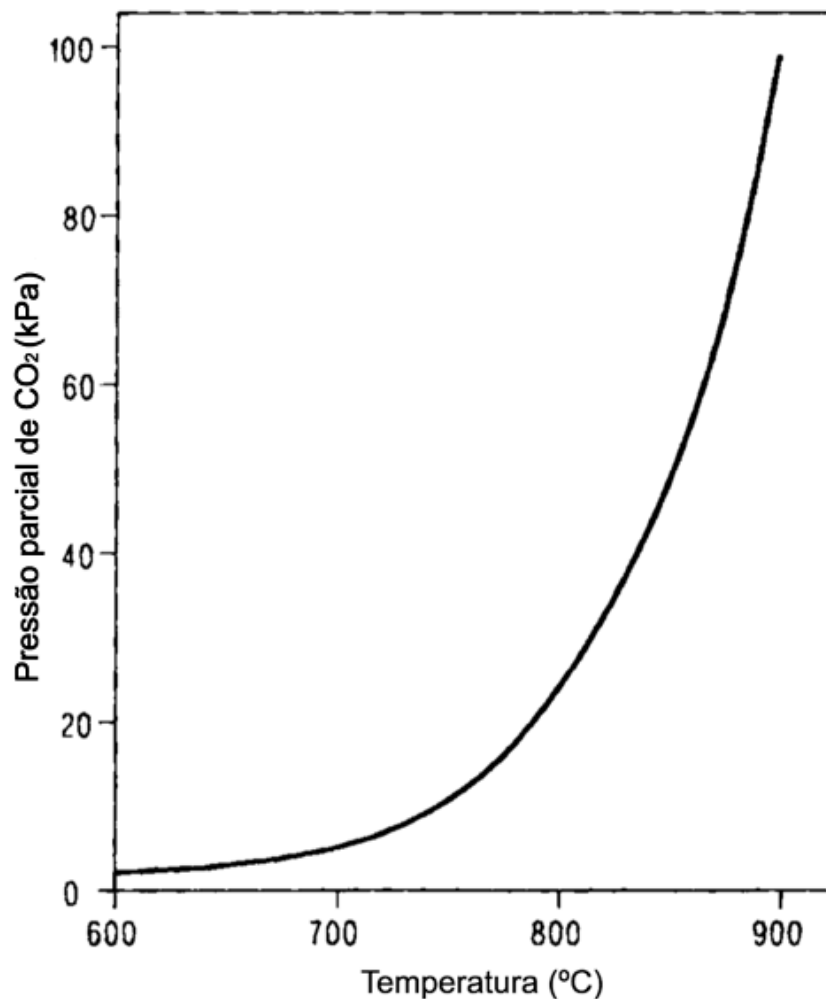


Figura 4.1 - Variação da pressão de dissociação com a temperatura (Oates, 1998).

O calor de dissociação da calcite a 25°C tem sido frequentemente referido como estando na gama de 2909,8 a 3491,8 kJ/kg de CaO. Boynton cita um

valor médio de 3223,8 kJ/kg, enquanto Schwarzkorf (1978), que publicou posteriormente a Boynton, defende um valor de 3156,8 kJ/kg (Oates, 1998). Já Oates, último a publicar, prefere um valor de 3181,96 kJ/kg.

O calor de dissociação a 900 °C, calculado a partir do valor acima de 3181,96 kJ/kg, é 3027,1 kJ/kg CaO (a diferença de 154,86 kJ/kg de CaO resulta do facto de que 1,786 kg de CaCO_3 requer 1804,5 kJ para ser aquecido de 25 °C a 900 °C, considerando que os produtos de reação – 1,0 kg de CaO e 0,786 kg de CO_2 – libertam 870,85 e 778,75 kJ respetivamente para o arrefecimento de 900 °C a 25 °C, com um total de 1649,6 kJ). Schwarzkorf (1978) reporta um valor de 2922,4 kJ/kg para 900 °C.

Nota-se que a reação é reversível.

4.6. – Cinética de calcinação

Como visto anteriormente, a passagem de uma partícula de calcário através de um forno pode ser dividida em cinco etapas. A seguinte descrição refere-se ao caso de uma pedra calcária de alto teor de cálcio.

a. Na zona de pré-aquecimento, o calcário é pré-aquecido desde a temperatura ambiente até cerca de 800 °C por gases de escape, isto é, os produtos gasosos da combustão mais o CO_2 libertado na calcinação e o excesso de ar;

b. A cerca de 800 °C, a pressão do CO_2 produzido pela dissociação de calcário é igual à pressão parcial do CO_2 nos gases de escape do forno. À medida que a temperatura da pedra calcária aumenta, a camada superior começa a decompor-se, de modo que, quando a temperatura da pedra atinge 900 °C a camada de cal pode ser 0,5 mm de espessura (corresponde a cerca de 5% do peso de uma partícula de 25 mm de cal);

c. Uma vez que a temperatura do calcário excede a “temperatura de decomposição” de 900 °C, a pressão parcial excede 1 atm e o processo de dissociação pode avançar para além da superfície da partícula;

d. Se todo o carbonato de cálcio se dissocia antes da partícula deixar a zona de calcinação, a cal começa a sintetizar. (Este processo ocorre de forma muito limitada durante a fase (c), mas para a maioria das situações pode ser desconsiderada);

e. A partícula de cal, que pode conter resíduos de calcário, sai da zona de calcinação a 900 °C e é arrefecida pelo ar posteriormente utilizado para a combustão.

As fases (a), (b) e (e) envolvem a transferência direta de calor entre o meio gasoso e a partícula. O mecanismo de transferência de calor é bem compreendido e não é específico para a calcinação de calcário. No entanto, as fases (c) e (d) são específicas da pedra calcária, pois são influenciados pela concepção do forno de cal e influenciam as propriedades da cal.

4.6.1. – Dissociação de calcário com alto teor de cálcio

A dissociação de calcário com uma temperatura superior à de decomposição pode ser considerada como consistindo em cinco processos, tal como ilustrado na Figura 4.2.

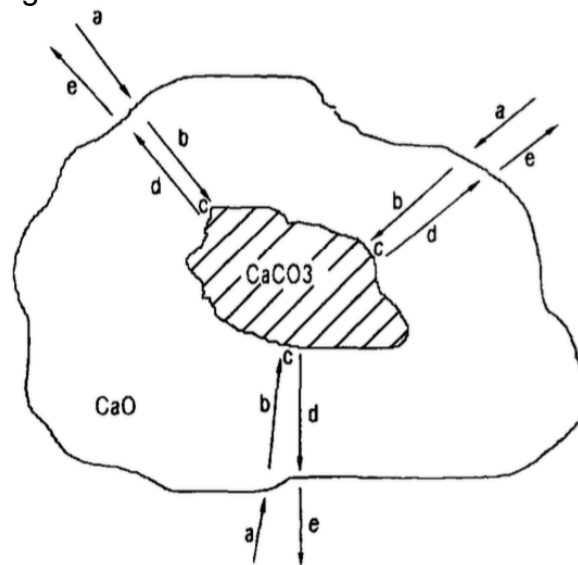


Figura 4.2 - Ilustração do processo de dissociação de calcário (Oates, 1998).

- a. O calor é transferido dos gases de escape para a superfície da partícula em dissociação;
- b. O calor é então conduzido a partir da superfície para a interface da reação através da camada microporosa da cal (ver Figura 4.2);
- c. Uma vez o calor transferido para a interface da reação ocorre a dissociação de CaCO_3 em CaO e CO_2 ;
- d. O CO_2 libertado na interface de reação, transfere-se através da camada de cal para a superfície da partícula e é aquecido pela corrente de gases de combustão;
- e. CO_2 sai da superfície para a fase gasosa.

A física e a química dos processos (a), (c) e (e) são de compreensão relativamente fácil, mas o efeito da microestrutura da camada de cal nos processos (b) e (d) é complexo e depende em parte nas características da pedra de calcário. Outras complicações surgem quando ocorrem alterações na

estrutura das camadas superfícies da cal, causadas por sinterização, escória e a absorção de dióxido de enxofre (Oates, 1998).

As generalizações acima são ilustradas pelos resultados de um trabalho laboratório sobre calcinação de uma esfera de calcário apresentado por Oates (1998), ver Figura 4.3. A esfera, suspensa num forno a 1200 °C, apresenta um diâmetro de 31 mm, tem um termopar embebido na camada superfície e outro 5 mm abaixo da superfície. A temperatura da superfície sobe acima dos 900 °C durante 5 minutos, e em seguida, aumenta progressivamente até 1000 °C após 60 minutos, e atinge a temperatura de operação (1200 °C) do forno após 67 minutos, altura em que a partícula fica totalmente calcinada. A calcinação da parte exterior da esfera, inferior a 5 mm, mantém a temperatura da superfície abaixo dos 900 °C durante 45 minutos, até que a interface da reação o atinga. A temperatura nessa altura sobe para a temperatura da superfície. A percentagem calculada de CaCO_3 que é dissociado e o raio do núcleo da pedra calcária também são ilustradas na Figura 4.3.

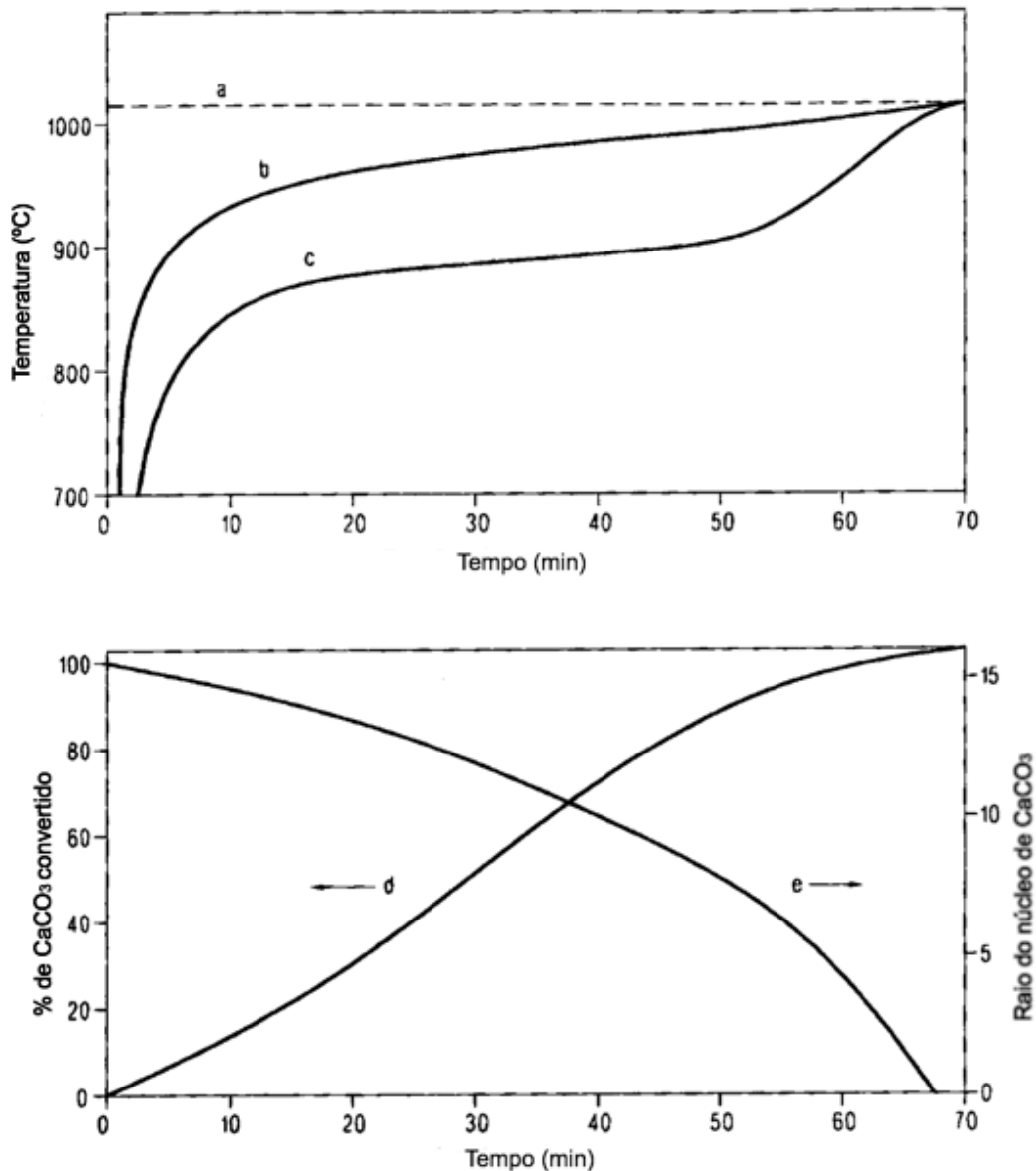


Figura 4.3 - Processo de calcinação de uma pedra calcária esférica com 31mm (Oates, 1998). (a) temperatura no interior do forno; (b) temperatura da superfície; (c) temperatura 5mm abaixo da superfície; (d) % de CaCO_3 dissociado; (e) raio do núcleo de CaCO_3 .

4.7. – Fatores que influenciam a calcinação

As principais variáveis que afetam a taxa de dissociação são: as propriedades da carga do forno; o desempenho físico-químico do forno; a distribuição e aproveitamento do combustível. Estes aspectos são agora sucintamente analisados.

4.7.1. – Propriedades da carga do forno

Uma condição básica para a fabricação de cal que atenda às melhores normas técnicas é o exame das características das rochas carbonáticas antes de serem carregadas no forno. É preciso identificar a sua composição química, composição mineralógica, cristalinidade, impurezas e granulometria, para se conseguir uma boa conduta da rocha no forno e um produto qualificado.

No interior do forno a pedra para calcinação pode apresentar defeitos que dificultam a operação, tais como:

- i. Tendência a crepitar o material, provocando uma pulverização incontrolável;
- ii. Tendência a distribuir-se mal, impedindo ou dificultando o contacto com gases quentes ao longo da extensão do forno. É algo que é bastante relatado no sector industrial na operação em fornos rotativos.

4.7.2. – Desempenho do forno/reação físico-química

No interior do forno, a rocha carbonática passa por diversas fases à medida que é aquecida. De acordo com o tipo de forno e as características da carga, como já abordado anteriormente, essas fases podem ser assim sintetizadas:

- i. *Pré-aquecimento*: as pedras são pré-aquecidas pelo contacto com os gases de combustão;
- ii. *Aquecimento*: feito segundo uma taxa de aquecimento adequada, até atingir a temperatura de decomposição;
- iii. *Calcinação*: quando é atingida a temperatura de decomposição, dá-se à carga um dado “tempo de residência” para que o calcário se transforme em cal da forma mais completa possível;
- iv. *Arrefecimento*: acontece na zona de descarga, geralmente por intermédio de ar insuflado, que serve também como ar secundário de combustão do forno.

O processo de transformação dos blocos de rocha componentes da carga começa nas partes mais quentes (em contacto com os gases), evoluindo como “ondas térmicas” em direção às zonas frias (interior da pedra). Como é a primeira camada a se formar no processo de calcinação e como possui condutividade térmica bem menor do que a rocha, a cal faz com que a velocidade de avanço da camada calcinada diminua à medida que se aproxima do núcleo da pedra submetida ao calor. Assim, de acordo com a disposição e a granulometria da carga no forno, além da distribuição de calor ao longo das

zonas de aquecimento e calcinação, é frequente encontrar no material calcinado uma zona central (núcleo) não decomposta, isto é, na forma de carbonato (Schwarzkorf, 1978). Na Figura 4.4 mostra uma pedra de cal que apresenta um núcleo de carbonato de cálcio não calcinado.



Figura 4.4 - Pedra de cal com núcleo de carbonato não dissociado (Quallical, 2015).

Azbe (1953) concluiu que para um dado fator de forma o tempo de calcinação é diretamente proporcional à espessura ao quadrado (onde a espessura é definida como a dimensão mínima medida através do centro de gravidade da partícula). Azbe (1953) também derivou uma fórmula para a taxa de calcinação de um formato de partícula. Na Figura 4.5 pode se ver a variação do tempo de calcinação com a temperatura e tamanho de pedras densas.

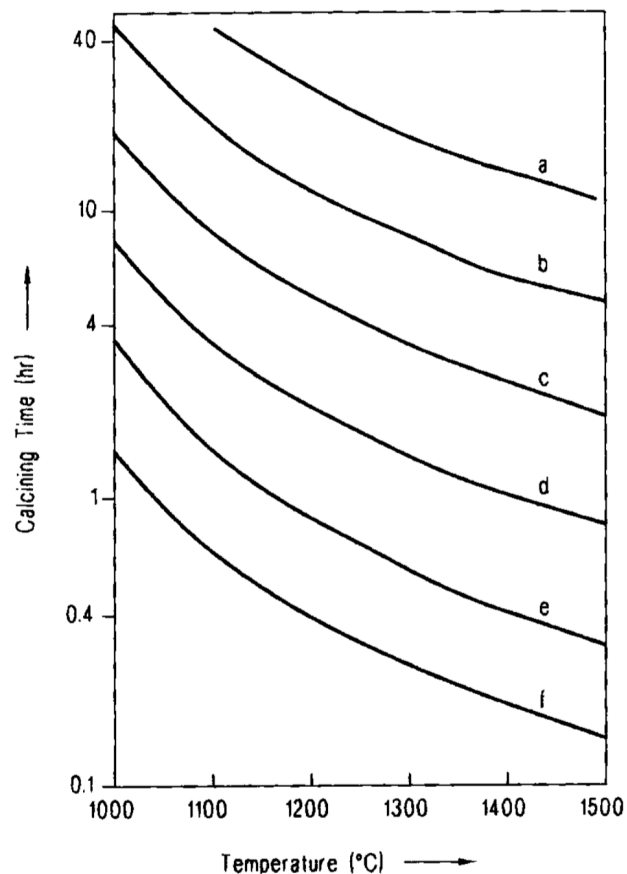


Figura 4.5 - Variação do tempo de calcinação com a temperatura e tamanho da pedra calcária. (Azbe, 1953).
 Pedra calcária com diâmetro de (a) 150mm; (b) 125mm; (c) 100mm; (d) 75mm; (e) 50mm; (f) 25mm.

A Figura 4.6, exemplifica o efeito de distribuição de zonas, descrito por Guimarães (2002) num forno vertical.

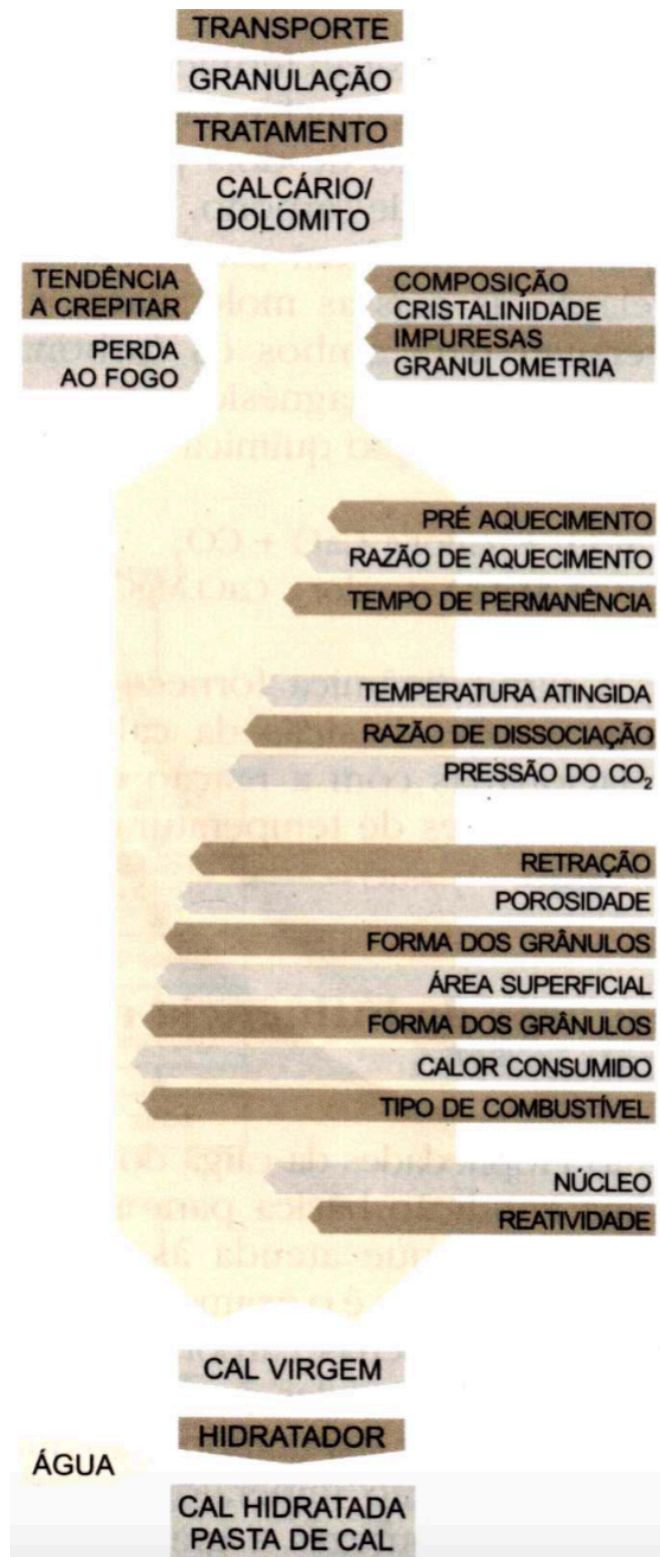


Figura 4.6 - Gráfico de distribuição de zonas no forno vertical (Guimarães, 2002).

Na Figura 4.6, o calcário na granulometria especificada ao processo é alimentado na parte superior do forno em contracorrente com a zona de queima. Na entrada do forno, devido ao choque térmico oferecido à pedra calcária, esta tem tendência a crepitar, com a geração de finos de minério. Em seguida o minério passa por uma região de pré-aquecimento. O tempo pelo

qual o minério não sofre qualquer reação de decomposição, estando sujeito unicamente ao aquecimento é função das características do minério (composição, cristalinidade, impurezas, granulometria).

A transformação do calcário em cal inicia-se quando a temperatura necessária para a reação de decomposição é atingida, este processo pode ocorrer ainda antes da zona de calcinação, ou seja, na zona de pré-aquecimento consoante o tempo de permanência do calcário nessa zona. Nessa zona inicia-se efetivamente o processo de calcinação do calcário. É importante salientar que fatores cinéticos, como a taxa de aquecimento e termodinâmicos, como a pressão parcial de CO_2 , são de fundamental importância na produtividade e qualidade do produto.

No decorrer da calcinação o material tem uma pequena tendência à retração de volume, ficando mais suscetível ao aumento de porosidade e área específica. Nesta etapa ocorre também a transformação dos retículos cristalinos referentes à formação dos óxidos. Isto é fases cristalinas referentes aos carbonatos vão gradativamente sendo substituídas pelas fases ligadas à presença dos óxidos.

Finalmente, na zona de descarga do forno o material tem as suas principais características definidas: reatividade, presença ou não de núcleo, existência ou não de supercalcinação. A importância do tempo de residência nas regiões finais do forno está relacionada com a sinterização do material ou ao fim do processo de calcinação. No caso da reação de decomposição não se completar, o material apresenta um núcleo não calcinado (Figura 4.4). Por outro lado, no caso de ocorrer supercalcinação, o óxido passa a sofrer sinterização da superfície da partícula de óxido (redução de porosidade). Esse tem características de cales duras, as quais apresentam alta pureza e dificuldade de hidratação.

Em suma, no projeto e operação do forno deve-se certificar que a decomposição térmica do calcário é finalizada e que o material não permanece no interior da zona de calcinação um tempo superior ao adequado, de modo a não sofrer sinterização.

A Figura 4.7 e a Figura 4.8 exibem respectivamente esquemas da transformação da estrutura das partículas de cal e micrografias eletrônicas que comprovam as transformações atrás referidas.

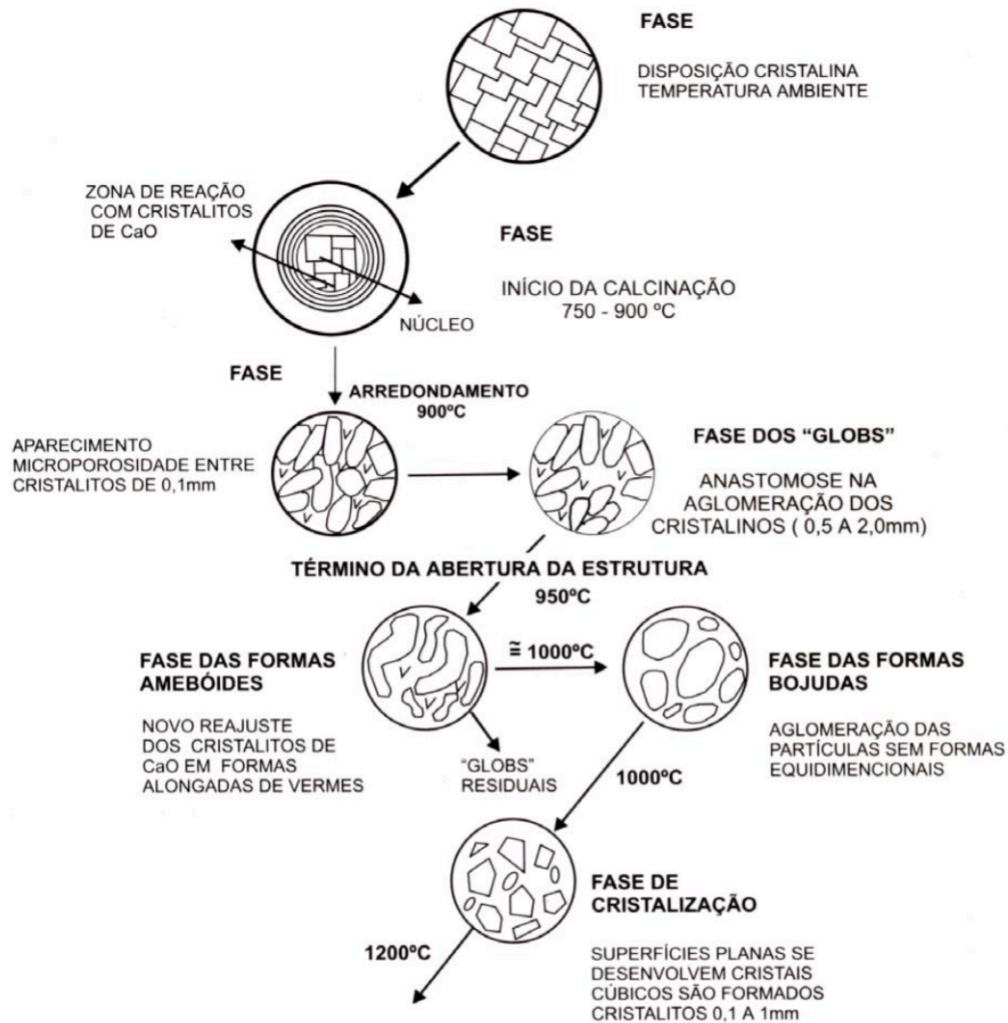


Figura 4.7 - Croqui da transformação da estrutura das partículas de cal (Guimarães, 2002).

Observa-se que para temperaturas entre 900 °C e 950 °C ocorre um processo de formação de poros no material e uma aglomeração de novos retículos cristalinos. Já a temperaturas superiores, entre 1000 °C e 1200 °C, as partículas aglomeram-se de forma mais concisa, reduzindo a microporosidade do material.



Figura 4.8 - Microscopia eletrônica de varredura de partículas de cal a diferentes condições de calcinação (Guimarães, 2002).

A calcinação perfeita das rochas calcárias depende particularmente das condições de operação do forno e da qualidade da matéria-prima. Dos fatores que dependem fundamentalmente da qualidade da matéria-prima, destaca-se a presença de impurezas, que diminui a percentagem de CaO libertado, inferindo, por conseguinte, a redução na reatividade da cal. O problema só é reduzido com calcinação a baixa temperatura. A presença de 2 % de impurezas, por exemplo, pode ocasionar a perda até 8 % de CaO livre. De

forma elementar, as impurezas têm reações diferentes: a sílica, o ferro e a alumina tendem a formar silicatos; o enxofre contamina o meio ambiente quando é expulso junto com os gases de escape, o óxido de sódio reduz a contração; fósforo, potássio e certos minerais valorizam a cal quando usada na agricultura, pois são nutrientes e micronutrientes de vegetais. Muitas vezes, elementos como fósforo, enxofre e boro impedem certas aplicações da cal.

Na constituição da cal salienta-se a importância da sílica, alumina e ferro, ao lado do cálcio e, mais subordinadamente, do magnésio. A Figura 4.9 mostra a influência da temperatura e presença de impurezas, nas diferentes fases cristalinas.

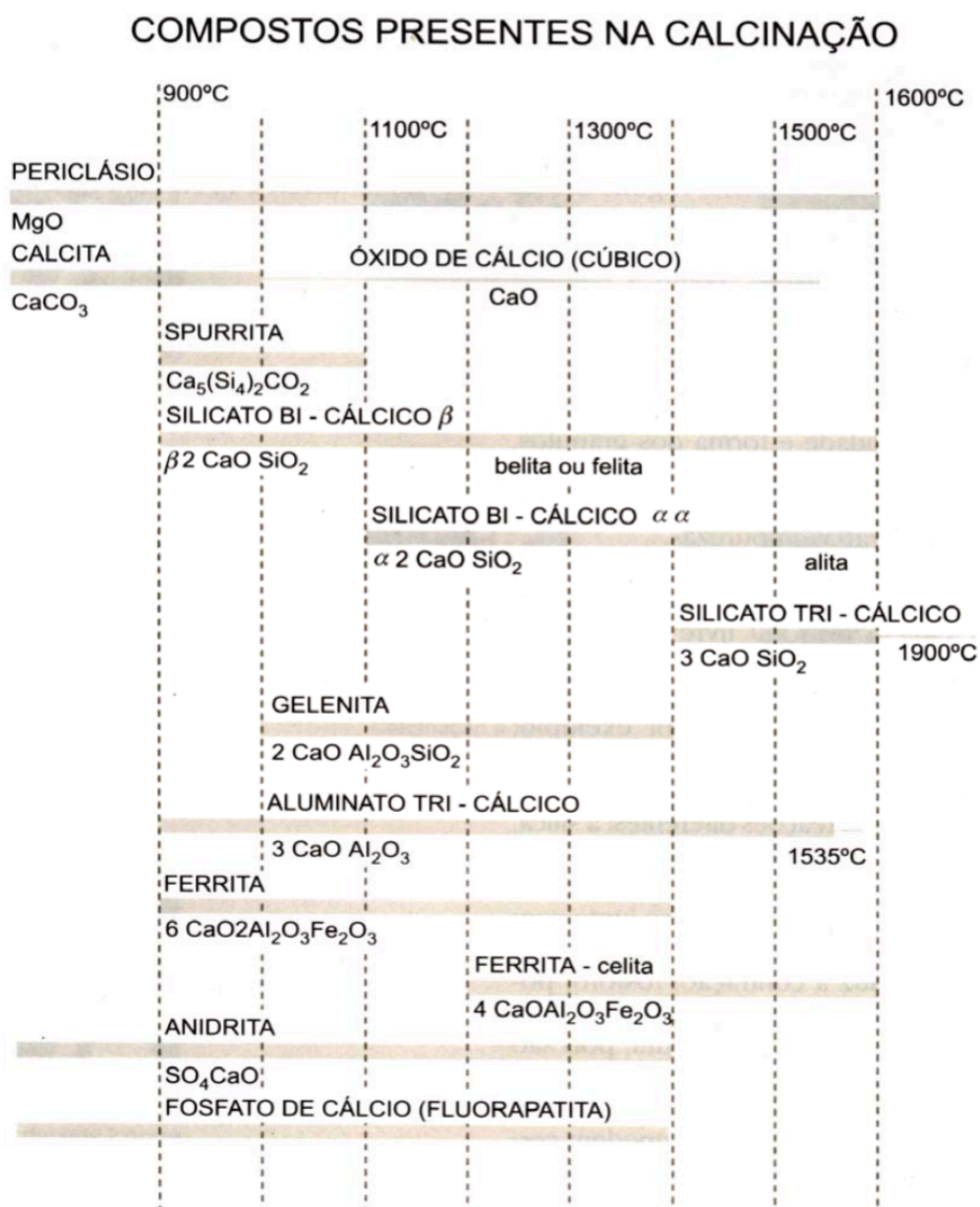


Figura 4.9 - Influência da temperatura de calcinação (Millet, 1979).

A qualidade química do produto depende, primeiro, das características e das impurezas contidas nas rochas que lhe dão origem. Assim, recebe o nome de cal calcítica o produto com alto teor em óxido de cálcio. A qualidade da cal depende também da operação do forno. Quando este é operado com habilidade, seja em relação ao controlo de temperatura e seja em relação ao tempo de residência, o forno de calcinação produz cal de alta reatividade, denominada popularmente de cal gorda, mole ou rica. Já num forno operado visando principalmente a maior produtividade, produz cal comumente chamada de cal dura, de alta densidade e baixa reatividade, no caso de temperaturas elevadas e menor tempo de residência. Ainda objetivando a alta produtividade, no caso de temperaturas e tempos de residência inferiores aos técnicos, ocorre a produção de cal virgem denominada de crua, de alto teor de CO_2 e baixa efetividade (Guimarães, 2002).

4.7.3. – Combustível, aproveitamento e distribuição no forno

O calor é o segundo componente da reação química que resulta na formação da cal. As fontes combustíveis exercem grande influência sobre a qualidade do produto e sobre os custos do processo de fabricação, estimulando também a evolução técnica dos fornos. Assim, a escolha do combustível e os cuidados com a recuperação de calor excedente no processo devem ser criteriosamente planejados.

A quantidade mínima de energia requerida no processo de dissociação térmica do calcário a temperaturas próximas de 1173 K (900 °C) é aproximadamente 3029 kJ por kg de CaO produzido. Já o calor de dissociação da reação na condição padrão de 298 K (25 °C) é de 3184 kJ/kg de CaO. O calor total empregue nos fornos de calcinação gira em torno de 4484 kJ/kg de CaO (Boynton, 1980; Oates, 1998).

O rendimento térmico de um forno de calcinação industrial é definido como a energia teórica requerida para a promoção da reação de calcinação dividida pela energia real fornecida ao forno. A perda térmica nos fornos de calcinação dá-se com a irradiação de calor pela carcaça do equipamento e entre outras, como pelos produtos de combustão expelidos pela chaminé; com o CO_2 produzido pela decomposição dos carbonatos; com poeiras e minerais incombustíveis que acompanham os gases; nos produtos (cal) descarregados com temperatura acima do ambiente. A redução dessas perdas pode ser conseguida com diversas ações: como a pintura do forno; revestimento de tijolo refratário com composição variável de acordo com o tipo e as zonas do

forno; capas de isolamento de calor; aumento da altura do forno em relação à zona de pré-aquecimento; uso de equipamento de controlo de temperatura, alimentação de carga e outros; ou mesmo adoção de um novo modelo de forno ou equipamento.

O rendimento térmico, que determina a quantidade de calor consumido na calcinação, também está ligado à participação das impurezas na cal e à tolerância à presença da pedra não dissociada (medida pelo teor de CO_2) na cal.

4.8. – Sinterização de cal com alto teor de cálcio

Quando um calcário calcítico denso é calcinado a temperaturas próximas de 900°C , o volume das partículas individuais aumenta muito ligeiramente até cerca de 60 % do calcário dissociado e, em seguida, diminui ligeiramente a cerca de 98 % do volume original, no ponto onde a calcinação é completa (Anderson, 1973). O processo de sinterização após a conclusão de dissociação é bastante importante para a prática de “queima de cal”, por causa do seu efeito sobre a reatividade da cal. A Figura 4.10 mostra o efeito da temperatura e do tempo de sinterização, medida por variação da densidade aparente, de uma pedra de cal produzida a partir de uma pedra calcária densa de alto teor de cálcio.

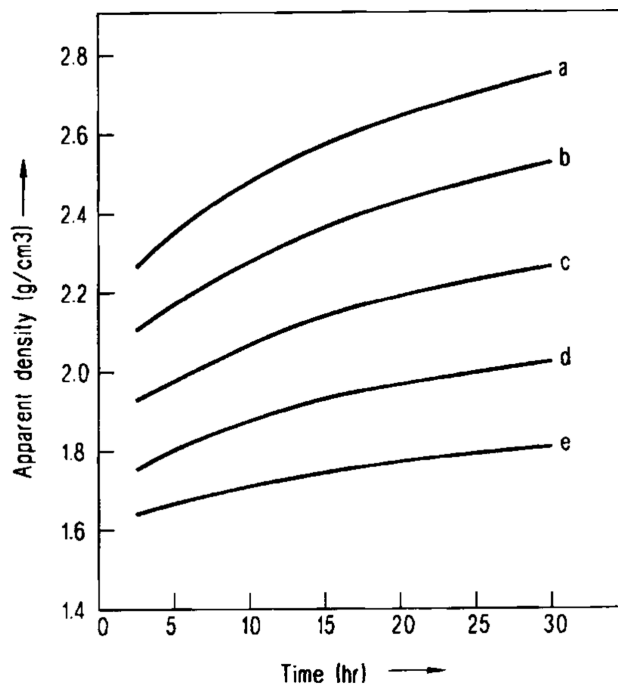


Figura 4.10 - Variação da massa volúmica aparente com a temperatura e tempo (Oates, 1998).

(a) 1400°C ; (b) 1300°C ; (c) 1200°C ; (d) 1100°C ; (e) 1000°C .

As micrografias eletrônicas da cal demonstram a mudança dramática na estrutura resultante do processo de sinterização (ver Figura 4.8).

Sabendo que a calcinação de uma pedra de calcário começa a partir do exterior em direção ao núcleo, Predescu (1989) defende que a sinterização irá avançar a uma velocidade constante de 5 mm por hora para dentro do núcleo. Tendo em conta que todas as pedras calcárias são perfeitamente esféricas, como foi assumido anteriormente, Predescu (1989) calculou as quantidades de calcário calcinado a cada hora apresentadas na Tabela 4.2.

Tabela 4.2 - Quantidade de calcário calcinado a cada hora (Predescu, 1989).

Após 1h	36,0 %	Após 6h	98,2 %
Após 2h	61,6 %	Após 7h	99,4 %
Após 3h	78,9 %	Após 8h	99,7 %
Após 4h	89,5 %	Após 9h	99,8 %
Após 5h	95,4 %	Após 10h	100 %

As implicações do processo de sinterização, juntamente com os efeitos do tamanho de partículas sobre o tempo de calcinação é demonstrado de seguida. Quando pedras calcárias de 50 mm a 150 mm, que consistem em pedras esféricas, são calcinadas num forno a 1200 °C os tempos necessários para a calcinação completa são os indicados na Tabela 4.3.

Tabela 4.3 - Valores estimados da dissociação e sinterização após 10 h a 1200 °C (Oates, 1998).

Diâmetro da pedra [mm]	Tempo de dissociação [h]	Propriedades esperadas ao fim de 10h	
		% CaCO_3 ¹	Massa volúmica [g/ml]
50	0,8	0	2,05
75	2,0	0	2,05
100	4,8	0	1,95
125	11	1,8	1,6
150	26	43	1,5

Se o tempo de calcinação for restrito a 10 h, as pedras com 50, 75 e 100 mm seriam inteiramente calcinadas. A pedra de 125 mm seria quase calcinada completamente. Já a pedra de 150 mm ainda contém mais de 40 % da massa de CaCO_3 e a massa volúmica da camada de cal seria de cerca de 1,5 g/ml.

¹ % por massa.

Capítulo 5

Modelo de cálculo

Neste capítulo irá ser mostrado todo o processo de cálculo efetuado para o dimensionamento do forno. O dimensionamento depende não só das dimensões físicas do forno, mas depende também das propriedades físicas do fluido e combustível para qual o sistema é dimensionado.

5.1. – Dimensão do forno

A caracterização de partículas é importante em todos os aspetos da produção de partículas, fabricação, manipulação, transformação e aplicação. A caracterização das partículas é a primeira etapa requerida num processo que envolva partículas sólidas. Esta caracterização inclui não só os parâmetros intrínsecos estáticos – tais como tamanho, densidade, forma e morfologia – mas também o seu comportamento dinâmico em relação ao escoamento do fluido – como o coeficiente de arrasto e velocidade terminal (Yang, 2003).

A caracterização de uma única partícula requer a medição e definição das características das partículas tais como tamanho, densidade, forma e morfologia da superfície. O tamanho da partícula é a dimensão linear mais apropriada para caracterizar uma partícula individualmente, uma partícula ideal como uma esfera é caracterizada, unicamente, pelo seu diâmetro (d_p). Tendo em conta o diâmetro da partícula é possível determinar o volume desta.

$$V_p = \frac{4 \pi}{3} \left(\frac{d_p}{2} \right)^3 \quad (5.1)$$

onde,

d_p : Diâmetro da partícula.

Sabendo a quantidade de CaCO_3 e a sua massa volúmica, determina-se o volume líquido da carga no interior do forno.

$$V_{liq} = \frac{\text{Massa de } \text{CaCO}_3}{\rho_{\text{CaCO}_3}} \quad (5.2)$$

Fornos verticais são, basicamente, reatores de leito embalado (ou de enchimento) com um fluxo ascendente de gases que passam por contracorrente relativamente ao fluxo descendente constituído por partículas sólidas.

Os leitos fixos, ou os leitos de enchimento, desempenham um papel vital em processos químicos. O problema operacional primário para um leito fixo é a queda de pressão através do leito fixo de sólidos, como é o caso em estudo. Assim, uma melhor compreensão de operação e projeto de um leito fixo requer um estudo das partículas “embaladas” e o seu efeito sobre a queda de pressão através do leito (Yang, 2003). Embora no caso presente o reator não seja propriamente em leito fixo, porque este se desloca na direção descendente por ação da gravidade, sendo por isso a sua designação mais correta de leito embalado ou de enchimento, frequentemente neste texto usa-se, por simplicidade de descrição, a designação de leito fixo.

A porosidade é definida como a fração de volume livre do leito, pode ser determinada a partir do volume do espaço vazio e o volume total do leito. Esta pode ser afetada pelo modelo de empilhamento, formato da partícula, distribuição do tamanho e rugosidade da superfície da partícula e também pela altura do leito (Zou *et al.*, 1995).

O enchimento aleatório de partículas esféricas de tamanho uniforme é complexo mas pode ser descrito matematicamente através de um número de coordenação. De acordo com Scott (1960), o enchimento aleatório solto é criado inclinando o recipiente horizontalmente e rodando lentamente sobre o seu eixo e, em seguida, repondo-o lentamente para a sua posição vertical. O enchimento aleatório denso pode ser desenvolvido vertendo as partículas no interior dos recipientes cilíndrico e, em seguida, agitando-o durante vários minutos (Yang, 2003).

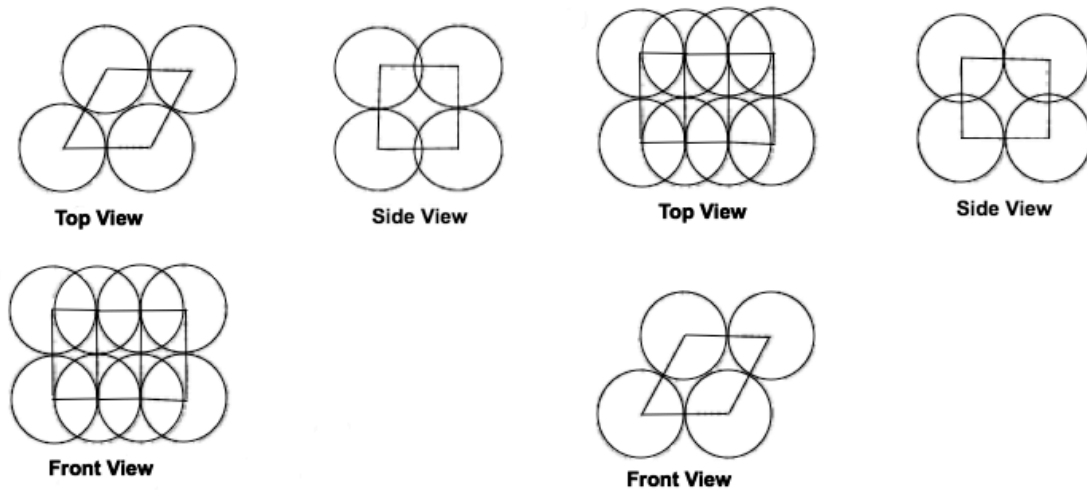


Figura 5.1 - Possíveis arranjos de esferas - dois ortorrômbo (Yang, 2003).

Para o empacotamento regular de partículas esféricas, a densidade de enchimento aleatório densa obtida varia de 0,61 a 0,63, equivalente a uma porosidade de 0,39 e 0,37, o que corresponde de perto ao empacotamento (arranjo) ortorrômbo com uma porosidade de 0,3955 (Yang, 2003).

Tabela 5.1 - Características de empacotamento (Yang, 2003).

Empacotamento	Pontos de contacto	Porosidade [%]
Ortorrômbo (duas orientações)	8	39,55

A área superficial específica do leito é definida como a relação da área de superfície total da partícula para o volume total do leito. Uma vez que o número de esferas por unidade de volume do leito é definido por (Yang, 2003):

$$N_p = \frac{6(1 - \varepsilon)}{\pi d_p^3} \quad (5.3)$$

sendo,

ε : Porosidade.

Conhecendo o volume líquido de ocupação pela carga no interior do forno e o número de partículas por unidade de volume é agora possível determinar o número de pedras no interior do forno. Este, pode então ser determinado por

$$n_{p_t} = N_p V_{liq} \quad (5.4)$$

Sabendo o número de pedras no interior do forno e o volume que cada partícula (pedra) possuir, é possível determinar o volume total de ocupação.

$$V = n_{p_t} V_p \quad (5.5)$$

Uma vez que o forno possui a característica geométrica de um cilindro, a altura deste é

$$H = \frac{V}{\pi r^2} \quad (5.6)$$

O cálculo do consumo de energia da zona de calcinação e arrefecimento têm de ser estudadas em conjunto. A zona de pré-aquecimento será analisada em separado. Essa divisão é necessária porque a temperatura do gás de escape (T_g) entre o pré-aquecimento e a zona de calcinação deve ser mais elevada do que a temperatura de equilíbrio do sólido, de modo que a Segunda Lei da Termodinâmica seja cumprida, isto é, que haja uma transferência de calor da corrente gasosa para a pedra, com uma diferença finita de temperaturas.

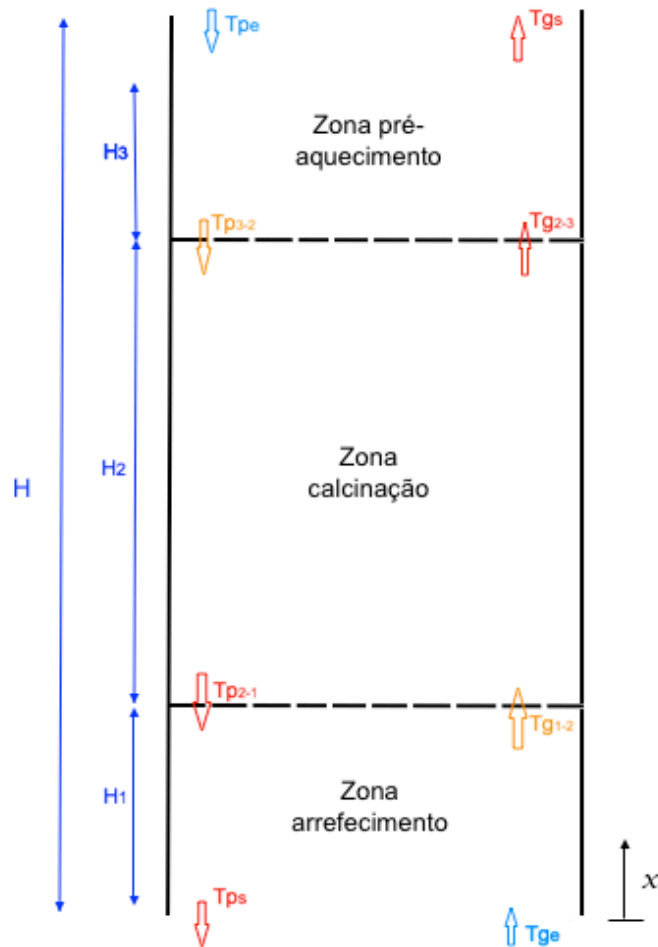


Figura 5.2 - Esquema da análise do forno.

5.1.1. – Dimensão da zona de calcinação

Sabendo o tempo de residência da carga na zona de calcinação e a quantidade de CaO pode-se facilmente determinar qual a quantidade de CaCO_3 na zona de calcinação. Sabe-se que 1 kg de CaCO_3 gera 0,56 kg de CaO, então:

$$\text{Quantidade de } \text{CaCO}_3 \text{ na calcinação} = \frac{\text{CaO} \cdot t}{24 \text{ horas} \cdot r_{cc}} \quad (5.7)$$

onde,

r_{cc} : Rácio cal – calcário, assume o valor 0,56 kg;

CaO: Massa de CaO.

Uma vez determinada a quantidade de CaCO_3 na zona de calcinação, determina-se o volume líquido da carga recorrendo à equação (5.2).

O número de pedras na zona de calcinação pode ser agora determinado, sabendo o volume líquido da zona de calcinação e recorrendo à equação (5.4).

O volume de ocupação da carga, equação (5.5).

Equação (5.6) sabe-se a altura da zona de calcinação.

5.1.2. – Dimensão da zona de arrefecimento

Área da seção recta do forno.

$$A_t = \frac{\pi d_p^2}{4} \quad (5.8)$$

Área da base do forno.

$$A_b = \pi r^2 \quad (5.9)$$

O número de Reynolds é a relação entre as forças de inércia (convectivas) promotoras do movimento e as forças de viscosidade que se opõem ao movimento.

$$Re = \frac{d_p U_f \rho_g}{\mu_g} \quad (5.10)$$

O caudal de gases (corrente de ar que entra pela parte inferior do forno) é dado por:

$$\dot{m}_g = \rho_g U_f A_b \quad (5.11)$$

O coeficiente de transferência de calor entre o fluido (ar) e as partículas é obtido através do número de Nusselt.

$$h = \frac{k_g \overline{Nu}}{d_p} \quad (5.12)$$

O número de Nusselt corresponde fisicamente a um gradiente de temperaturas adimensional junto à partícula, e representa o coeficiente de convecção adimensional. Sendo assim o número de Nusselt é uma função da geometria do escoamento (Re), do fluido Prandtl (Pr) (Yang, 2003).

$$\overline{Nu} = \left(2 + 1,1 Pr^{\frac{1}{3}} Re^{0,6} \right) \quad (5.13)$$

Análise na secção do forno.

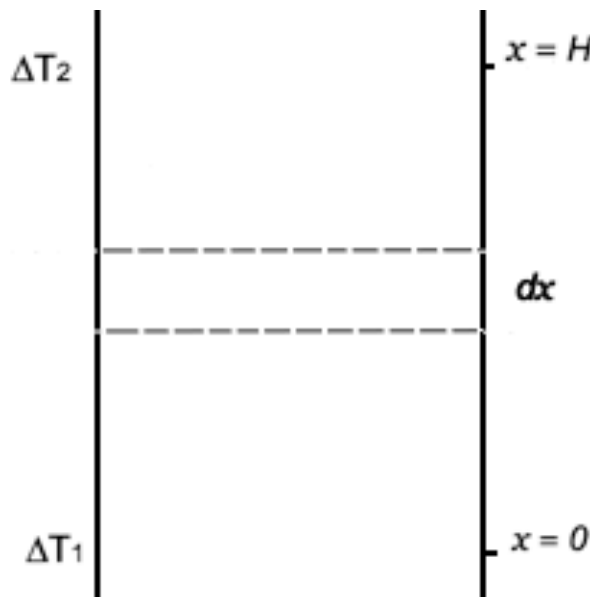


Figura 5.3 - Esquema da análise de um seção do forno.

Área de transferência por unidade de volume de leito:

$$\vartheta = \frac{6 (1 - \varepsilon)}{d_p} \quad (5.14)$$

A quantidade de energia em jogo na zona de arrefecimento da cal é dada por:

$$\dot{Q}_g = [\dot{m}_g c_p (T_{g_{1-2}} - T_{g_e})] \quad (5.15)$$

sendo,

\dot{m}_g : Caudal de ar;

c_p : Calor específico a pressão constante do fluido;

$T_{g_{1-2}}$: Temperaturas dos gases à entrada da zona de calcinação;

T_{g_e} : Temperaturas do fluido à entrada da zona de arrefecimento.

Sabendo a quantidade de energia consegue-se obter a altura da zona de arrefecimento (H_1):

$$\dot{Q} = \frac{h A_t 6 (1 - \varepsilon) H}{d_p} \frac{\Delta T_2 - \Delta T_1}{\ln \left(\frac{\Delta T_2}{\Delta T_1} \right)} \quad (5.16)$$

onde,

$$\Delta T_2 = T_{p_{2-1}} - T_{g_{1-2}} ;$$

$$\Delta T_1 = T_{p_s} - T_{g_e} .$$

5.1.3. – Combustão

Normalmente os combustíveis são classificados de acordo com a fase na qual são normalmente manuseados (Combustíveis sólidos, líquidos e gasosos). Do ponto de vista químico são os combustíveis gasosos os mais simples, como a gás natural, combustível selecionado para a simulação em estudo.

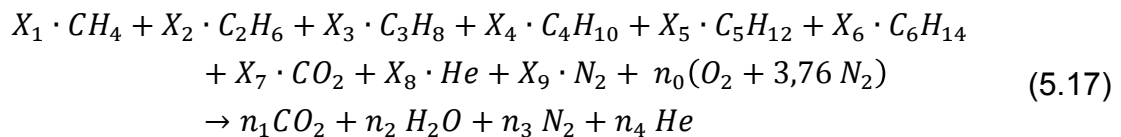
A composição dos combustíveis gasosos (misturas) é dada, normalmente, na base molar, ver Tabela 5.2.

O gás natural tem como componentes principais o metano (CH_4) e o etano (C_2H_6) variando gradualmente de composição consoante a origem.

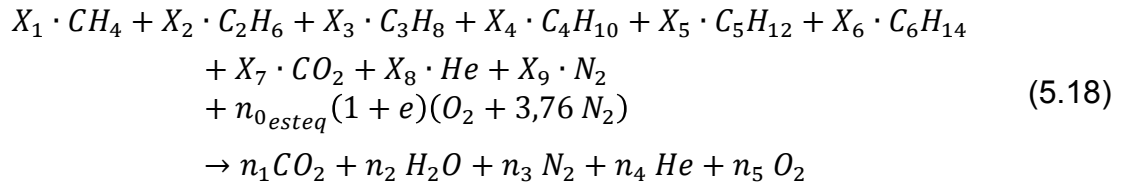
Tabela 5.2 - Composição típica de um gás natural (Trangás)

	Xi	Mi [g/mol]	Xi Mi
CH₄	0,837	16	13,39
C₂H₆	0,076	30	2,28
C₃H₈	0,0192	44	0,84
C₄H₁₀	0,007	58	0,41
C₅H₁₂	0,0017	72	0,12
C₆H₁₄	0,0008	86	0,07
CO₂	0,0023	86	0,10
He	0,002	4	0,01
N₂	0,054	28,15	1,52

Uma mistura reativa é denominada estequiométrica quando a quantidade de oxidante nessa mistura é a teoricamente necessária para queimar completamente todo o combustível num processo de combustão ideal. Na reação estequiométrica:

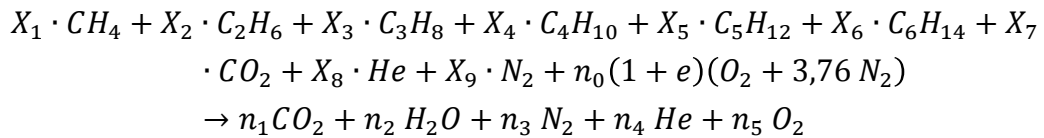


Na prática sabe-se que a combustão completa não é conseguida a menos que a quantidade de ar seja superior à teoricamente necessária (excesso de ar). Isto porque as características do queimador e do escoamento, não permitem a concretização de uma mistura ar – combustível perfeitamente homogénea. Ou seja a reação química acima referida (Equação 5.17), representa o processo de combustão ideal, sendo assim apenas uma aproximação do processo de combustão real. Para garantir a combustão completa do combustível, os equipamentos de combustão operam com excesso de ar.



Para a equação química anterior tem que se efetuar o balanço dos elementos químicos em jogo:

- Carbono: $n_1 = n_{1_{esteq}}$
- Hidrogénio: $n_2 = n_{2_{esteq}}$
- Oxigénio: $2 X_7 + 2 n_{0_{esteq}} (1 + e) = 2 n_1 + n_2 + 2 n_5$
- Azoto: $X_9 + 3,76 n_{0_{esteq}} (1 + e) = n_3$
- Hélio: $n_4 = X_8$



5.1.3.1. – Produtos

Tabela 5.3 – Composição dos produtos da combustão.

	M_i [kg/kmol]	c_{pi} [kJ/kg]
CO ₂	44	1,3197
H ₂ O	18	2,6042
N ₂	28,15	1,241
He	4	5,234
O ₂	32	1,2083

Fração molar dos produtos:

$$Xi = \frac{n_i}{\sum n_i} \tag{5.19}$$

Massa dos produtos:

$$m_p = n_1 (12 + 2 \cdot 16) + n_2 (2 + 16) + 28,15 n_3 + 4 n_4 + 2 \cdot 16 n_5 \quad (5.20)$$

Razão produto – combustível:

$$\text{Razão prod/comb} = \frac{m_p}{\sum X_i \cdot M_i} \quad (5.21)$$

Partindo do conhecimento da composição molar do combustível, calcula-se para cada componente da mistura a respetiva fração mássica Y_i :

$$Y_i = \frac{X_i \cdot M_i}{\sum X_i \cdot M_i} \quad (5.22)$$

O calor específico a pressão constante dos produtos de combustão virá dado por:

$$c_{p_p} = \sum Y_i \cdot c_{p_i} = Y_{10}c_{p_{10}} + Y_{11}c_{p_{11}} + Y_{12}c_{p_{12}} + Y_{13}c_{p_{13}} + Y_{14}c_{p_{14}} \quad (5.23)$$

onde,

$c_{p_{10}}$: Calor específico a pressão constante do CO_2 ;

$c_{p_{11}}$: Calor específico a pressão constante do H_2O ;

$c_{p_{12}}$: Calor específico a pressão constante do N_2 ;

$c_{p_{13}}$: Calor específico a pressão constante do He;

$c_{p_{14}}$: Calor específico a pressão constante do O_2 .

As propriedades físicas da mistura não são conhecidas com rigor. No entanto, é possível recorrer-se às propriedades dos gases componentes da mistura para se obterem valores globais.

A expressão seguinte permite o cálculo da viscosidade dinâmica de uma mistura de gases perfeitos (Pinho, 2011).

$$\mu = \sum_{i=1}^n \frac{X_i \cdot \mu_i}{\sum_{j=1}^n X_j \cdot \phi_{ij}} \quad (5.24)$$

em que,

$$\phi_{ij} = \frac{1}{\sqrt{8}} \left(1 + \frac{M_i}{M_j}\right)^{-\frac{1}{2}} \left[1 + \left(\frac{\mu_i}{\mu_j}\right)^{\frac{1}{2}} \cdot \left(\frac{M_j}{M_i}\right)^{\frac{1}{4}}\right]^2 \quad (5.25)$$

Para a condutibilidade térmica (Pinho, 2011),

$$k = \sum_{i=1}^n \frac{X_i \cdot k_i}{\sum_{j=1}^n X_j \cdot \phi_{ij}} \quad (5.26)$$

em que,

$$\phi_{ij} = \frac{1}{\sqrt{8}} \left(1 + \frac{M_i}{M_j}\right)^{-\frac{1}{2}} \left[1 + \left(\frac{k_i}{k_j}\right)^{\frac{1}{2}} \cdot \left(\frac{M_j}{M_i}\right)^{\frac{1}{4}}\right]^2 \quad (5.27)$$

5.1.3.2. – *Propriedades da combustão*

Calor necessário para dissociar uma quantidade de CaO:

$$\theta_{nec} = \theta_{reação} \cdot CaO \quad (5.28)$$

onde,

CaO: Massa de CaO.

Sabendo que o forno tem de produzir uma certa quantidade de CaO por dia (24 horas), o caudal de combustível para satisfazer a demanda é:

$$\dot{m}_{comb} = \frac{\theta_{nec}/PCI}{r_t} \quad (5.29)$$

Onde,

r_t : parâmetro de tempo, assumido como 86400 s.

Caudal dos produtos:

$$\dot{m}_p = Razão\ Prod/Comb \cdot \dot{m}_{comb} \quad (5.30)$$

5.1.4. – *Dimensão da zona de pré-aquecimento.*

Sabendo que 1 kg de $CaCO_3$ liberta 0,44 kg de CO_2 . O caudal de CO_2 [kg/s] libertado durante o processo de calcinação é:

$$\dot{m}_{CO_2} = \frac{CaCO_3 \cdot r_{cco_2}}{r_t} \quad (5.31)$$

onde,

r_{cco_2} : Rácio dióxido carbono – calcário, assumido como 0,44 kg;

CaCO_3 : Massa de CaCO_3 .

O caudal total de gases na zona de pré-aquecimento é dado por:

$$\dot{m}_{\text{pré-aque}} = \dot{m}_{g_{\text{pré-aque}}} + \dot{m}_{\text{CO}_2} + \dot{m}_p \quad (5.32)$$

sendo que o caudal do ar de arrefecimento das partículas de cal pode ser obtido através da Equação (5.11).

O coeficiente de transferência de calor por convecção nesta zona pode ser obtido através da Equação (5.12), o número de Reynolds através da Equação (5.10) e o número de Nusselt através da Equação (5.13).

A quantidade de energia é obtida pela Equação (5.15), assumindo o caudal total de gases. Assim, é possível calcular a altura da zona de pré-aquecimento através da Equação (5.16).

5.1.5. – Altura do forno

O somatório das alturas das zonas deve cumprir a seguinte condição para que seja possível produzir a quantidade de CaO desejado.

$$H_{\text{zona}} < H \quad (5.33)$$

onde,

$$H_{\text{zona}} = H_c + H_{\text{arref}} + H_{\text{pré-aque}} \quad (5.34)$$

5.3 – Transferência de calor na calcinação

Os gases quentes provenientes da queima do combustível trocam calor por convecção e radiação com a face interna da parede do forno. Da face interior da parede do forno a energia térmica é transferida por condução até à parede externa e esta, por sua vez, transfere o calor ao ambiente que cerca o forno por convecção.

O isolamento térmico do forno tem como principal objetivo isolar as paredes do forno de maneira a diminuir as perdas de calor trocadas com o meio ambiente de forma a aumentar o rendimento do sistema. Um objetivo secundário do isolamento é o de proteger pessoas e equipamentos das altas temperaturas existentes à superfície externa da parede. Com isto e sabendo que as temperaturas de operação do forno podem atingir valores até 1200 °C, o isolamento térmico tem um papel fundamental no que diz respeito à transferência de calor.

A resistência térmica à transferência de calor entre dois pontos do espaço determina-se a partir da Lei de Fourier, apresentada na Equação 5.35.

$$\dot{q} = -k A \frac{\partial T}{\partial r} \quad (5.35)$$

A transferência de calor por condução ocorre devido ao potencial gerado pela diferença de temperaturas entre dois pontos. A partir desse conceito de potencial de transferência de calor pode-se definir também o conceito de resistência a essa transferência, ou seja, o conceito de resistência térmica à transferência de calor.

Já a taxa de transferência de calor por convecção entre um volume sólido e um fluido pode ser escrita conforme a Equação (5.36):

$$\dot{q} = h A (T - T_{fluido}) \quad (5.36)$$

Pode-se perceber, novamente, que a transferência de calor ocorre devido à existência de um potencial, neste caso o potencial gerado pela diferença de temperaturas entre o volume sólido e o fluido.

A convecção no interior do forno (entre o gás e a parede), obtém-se tendo em conta a correlação Yang (2013).

$$h_{int} = 0,525 \cdot Re_c^{0,75} \quad (5.37)$$

No exterior do forno existe convecção natural, devido à ocorrência de uma diferença de temperaturas. O número de Grashof compara as forças ascensionais resultantes de uma diferença de massa volúmica com a variação da temperatura com as forças de viscosidade e é dado por (Incropera *et al.*, 2013):

$$Gr = \frac{g \cdot \beta \cdot (T_{par} - T_{amb}) \cdot L_c^3}{\nu^2} \quad (5.38)$$

Sendo por sua vez o número de Rayleigh definido como

$$Ra = Gr \cdot Pr_{amb} \quad (5.39)$$

As correlações válidas para uma placa vertical, são aplicadas neste caso à parede exterior do forno que é cilíndrica e vertical, desde que a espessura da camada limite seja muito menor que o diâmetro do cilindro (Incropera *et al.*, 2013). Para que essa condição se verifique é necessário que:

$$D \geq \frac{35 L_c}{Gr^{1/4}} \quad (5.40)$$

O número de Nusselt por convecção natural no exterior de uma parede é dado pela equação (5.41), a qual é adequada para a maioria dos cálculos em engenharia (Incropera *et al.*, 2013):

$$\overline{Nu} = \left(0,825 + \frac{0,387 \cdot Ra^{\frac{1}{6}}}{\left[1 + \left(\frac{0,492}{Pr_{amb}} \right)^{\frac{9}{16}} \right]^{\frac{8}{27}}} \right)^2 \quad (5.41)$$

O coeficiente de transferência de calor no exterior é dado por:

$$h_{amb} = \frac{\overline{Nu} \cdot k_{amb}}{L_c} \quad (5.42)$$

O coeficiente de transferência de calor efetivo é a medida da resistência global de uma série de barreiras convectivas e radiativas.

$$h_{ef} = h_{amb} (T_{par} - T_{amb}) + F_{1-1} \epsilon \sigma (T_{par}^4 - T_{amb}^4) \quad (5.43)$$

onde,

F_{1-1} : Fator de forma, assumido como igual à unidade.

Em geral, o processo de transferência de calor é dado por uma combinação de fenômenos. Assim, o fluxo de calor total transmitido num processo real pode ser escrito da seguinte forma:

$$q = \frac{T_{int} - T_{amb}}{R_{conv_{int}} + R_{cond_{ref}} + R_{cond_{iso}} + R_{conv_{ext}}} \quad (5.44)$$

5.4 – Perda de carga

A queda de pressão através do leito fixo é devida não só à resistência de atrito na superfície da partícula, mas também à expansão e contração do escoamento gasoso através dos interstícios existentes entre as partículas que constituem o leito de enchimento. A perda de carga é dada por (Yang, 2013):

$$\Delta P_i = \left[\left(\frac{17,3}{Re_i} + 0,336 \right) \cdot \left(\frac{\rho_{gi} U_{fi} H_i}{d_p} \right) \cdot (1 - \varepsilon) \cdot \varepsilon^{-4,8} \right] \cdot \alpha \quad (5.45)$$

onde α representa a majoração entre 20% a 30%, devido ao empilhamento da carga (Levenspiel, 1998).

A perda de carga total trata-se do somatório das perdas de carga de cada zona:

$$\Delta P = \sum \Delta P_i \quad (5.46)$$

O sistema do forno em estudo possui um ventilador para o escoamento dos gases, este ventilador necessita de uma potência que vença a perda de carga, bem como a possibilidade de funcionamento a altas temperaturas.

5.5 – Calcinação

Embora a ausência de informações e conhecimento aprofundados na área de Engenharia Química, impeça que seja analisada a taxa de calcinação e de sinterização com sucesso. É possível, no entanto, que valores referentes ao processo de calcinação sejam apresentados.

A reação de decomposição térmica do carbonato de cálcio tem como produtos finais o óxido de cálcio e o CO₂. Assim sendo, a pressão parcial do gás carbônico é uma variável que exerce influência significativa na decomposição térmica do calcário. Pois, a reação só ocorre que a pressão parcial de CO₂ no gás acima da superfície do calcário for menor do que a pressão de decomposição do próprio CaCO₃. Em suma, a presença de CO₂ inibe a reação de calcinação.

A pressão de equilíbrio do CO₂ a partir da decomposição de carbonato de cálcio em atmosferas (Silcox *et al.*, 1989) pode ser expressa por:

$$P_{eq} = 4,137 \times 10^7 \cdot \exp\left(-\frac{20474}{T_{int}}\right) \quad (5.47)$$

Havendo um aumento da pressão parcial de CO₂ na atmosfera que envolve a partícula a calcinar, a temperatura necessária para que se promova a reação de decomposição térmica do calcário é maior (García-Labiano 2002; Meier, 2004). O fator pré-exponencial (*Kd*) foi estimado através da Equação (5.48) por Silcox *et al.*, (1989), onde *Kd* o fator pré-exponencial da taxa de decomposição do calcário vem em mol/(m atm s) e *T_{int}* a temperatura de calcinação em Kelvin.

$$Kd = 1,22 \cdot \exp\left(-\frac{4026}{T_{int}}\right) \quad (5.48)$$

A taxa de calcinação da calcário é dada pela Equação (5.49) (Silcox *et al.*, 1989), estando R_c a taxa da reação de calcinação do calcário em mol/(m²s).

$$R_c = K_d \cdot (P_{eq} - P_{int}) \quad (5.49)$$

Capítulo 6

Programa de simulação computacional

Um dos principais objetivos desta dissertação é fornecer subsídios para o estudo da produção de cal em forno verticais. Para este fim foi desenvolvido um programa de simulação computacional de fornos verticais, que possibilita o estudo da influência dos principais parâmetros de projeto e de funcionamento deste tipo de forno. O programa de simulação computacional foi desenvolvido como ferramenta de auxílio no projeto de fornos verticais. As principais características de funcionamento e de utilização são presentes neste capítulo.

6.1. – Construção

Uma vez apresentada a metodologia de dimensionamento de um forno vertical de produção de cal, foi possível desenvolver um programa capaz de dimensionar um forno de uma forma autónoma e rápida. O programa em questão foi desenvolvido através de uma ferramenta informática denominada *NetBeans Java 8.1*.

O programa então desenvolvido é constituído por duas partes diferentes, embora intrinsecamente interligadas. A primeira parte do mesmo representa o código do programa em si, onde foram introduzidos todos os dados (valores constantes) e todas as equações necessárias para o procedimento de cálculo de dimensionamento de um forno, sendo que para tal foram geradas variáveis que possibilitam a elaboração do algoritmo de cálculo. A segunda parte do programa representa as duas interfaces que foram criadas, uma inicial onde o utilizador introduz os restantes dados para o cálculo e por último uma interface onde se apresentam os resultados finais. Deste modo o utilizador pode interagir com o programa de uma maneira intuitiva e eficaz. Para tal, é

necessário cumprir um rigor de etapas para a elaboração do programa. Na figura 6.1 sintetizam-se os procedimentos do programa.

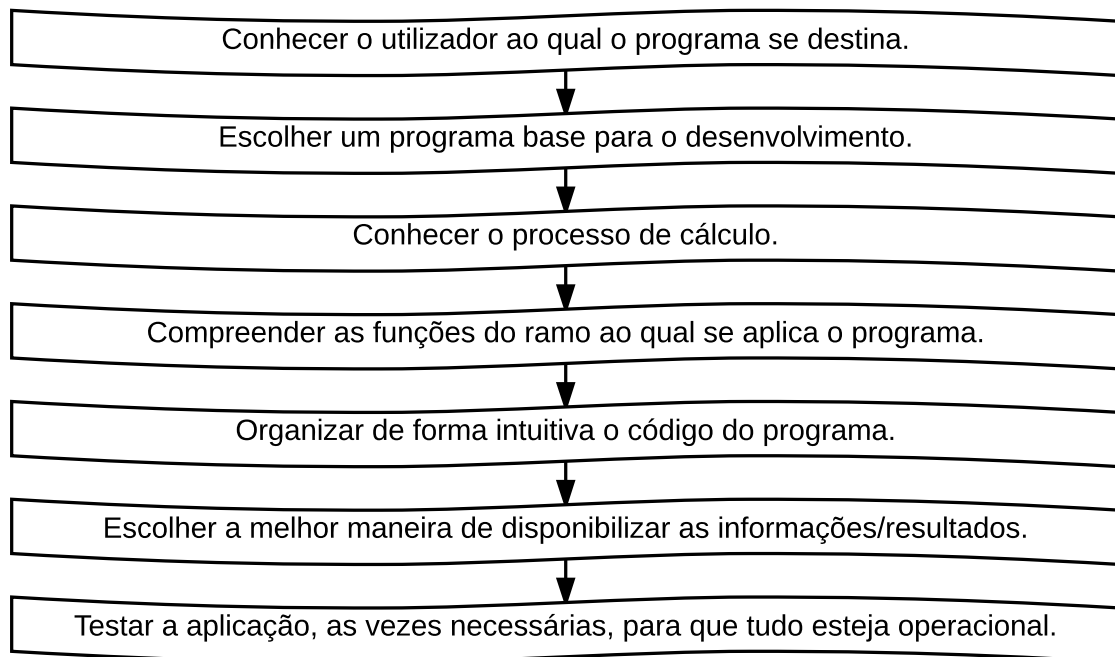


Figura 6.1 - Síntese da elaboração do programa.

6.2. – Programa base

O processo de cálculo para o dimensionamento possui métodos iterativos bem como processos de atribuição de variáveis que têm de ter a capacidade de tomar valores dependendo de certos parâmetros, várias ferramentas podiam ser escolhidas para este trabalho, tais como o MatLab, EES, Microsoft Visual Basic ou Java.

A ferramenta *NetBeans Java 8.1*. foi a escolhida não porque possuía uma maior capacidade em modelos matemáticos, mas porque já havia um conhecimento prévio desta ferramenta, o que torna assim mais acessível o desenvolvimento. Esta ferramenta possui um ambiente de desenvolvimento integrado gráfico, o que facilita a construção das interfaces da aplicação.

Tendo em consideração que não se trata da investigação de novos modelos de cálculo, mas sim de apoiar quem já os utiliza, foi criado um programa de raiz, com a capacidade de ajudar no dimensionamento de fornos

de cal verticais, tornando assim mais produtivo todo o processo e trabalho necessários em volta do assunto. Sendo no final do trabalho apresentado uma versão executável da aplicação.

6.3. – Apresentação do programa

O presente momento o programa de simulação é capaz de avaliar a influência das seguintes variáveis de projeto e de operação do forno:

- Geometria do forno (altura, raio interno e raio externo);
- Propriedades da carga (propriedades físicas, empilhamento das partículas e temperatura de alimentação);
- Parâmetros operacionais do forno (taxa de produção, tempo de residência da carga);
- Parâmetros operacionais da queima (tipo e consumo de combustível, excesso de ar de combustão, temperatura de admissão de ar);
- Isolamento das paredes do forno (propriedades físicas, espessuras das paredes).

As características de funcionamento do forno tornam a descrição matemática dos fenómenos de transferência de calor no interior do forno uma tarefa bastante complexa e exigem a adoção de algumas hipóteses simplificativas que venham a facilitar este processo de descrição. O modelo matemático desenvolvido para o programa de simulação considera as seguintes condições durante a realização dos balanços de massa e térmico:

- Tempo de permanência da carga na camada ativa de 10 horas;
- Transferência de calor por radiação entre volumes de controlo através de áreas superficiais muito pequenas, desprezável;
- Propriedades físicas dos materiais, tais como emissividade, densidade, calor específico, condutividade térmica e viscosidade dinâmica (excepto na zona de calcinação), constantes ao longo de todo o domínio computacional de cada zona;
- Transferência de calor por convecção entre as partículas e os gases;
- Transferência de calor por convecção entre a corrente gasosa e a parede interior do forno;

- Transferência de calor por convecção e radiação entre a face externa da parede e o ambiente externo ao forno.

Desta forma a modelação matemática desenvolvida para o programa de simulação consegue descrever de forma bastante completa os processos, sem, no entanto, dar origem a um sistema de equações demasiadamente complexo ou com não-linearidades muito intensas.

Como referido anteriormente, o programa é constituído por duas principais interfaces, onde a primeira representa numa janela de entrada de valores na qual o utilizador preenche as informações necessárias para o dimensionamento e a segunda apresenta os resultados finais do dimensionamento.

As figuras que se seguem referem-se à constituição gráfica das interfaces do programa.

The image shows a software window titled "FornoCal". It contains several input fields and a button. The inputs are: "Quantidade diária de CaO a produzir" (kg), "Raio do Forno (interno)" (m, dropdown menu showing 1), "Diâmetro da Pedra de CaCO3" (mm, with a note "(preferível usar valores médios)"), "Temperatura do meio ambiente" (°C), "Temperatura do ar à entrada do forno" (°C), "Temperatura da pedra de CaO à saída" (°C), "Combustível" (dropdown menu showing "Gas Natural"), and "Excesso de ar na combustão" (%). A "Calcular" button is at the bottom.

Figura 6.2 - Interface gráfica da entrada de valores.

The screenshot shows a software window titled 'FornoCal - Resultados'. It contains a list of calculated parameters for a vertical lime kiln, each with a text input field for the value and a unit. The parameters are: 'Altura do Forno' (12.726 m), 'Zona de arrefecimento' (1.227 m), 'Zona de calcinação' (5.303 m), 'Zona de pré-aquecimento' (0.683 m), 'Diâmetro do Forno (externo)' (3.6 m), 'Espessura da parede' (0.8 m), 'Temperatura da parede externa (Zona de calcinação)' (117.725 °C), 'Temperatura da parede externa (Zona de pré-aquecimento)' (89.22 °C), 'Caudal de Combustível' (0.012 kg/s), and 'Perda de carga' (927.875 Pa). At the bottom center is a button labeled 'Imprimir'.

Parâmetro	Valor	Unidade
Altura do Forno	12.726	m
Zona de arrefecimento	1.227	m
Zona de calcinação	5.303	m
Zona de pré-aquecimento	0.683	m
Diâmetro do Forno (externo)	3.6	m
Espessura da parede	0.8	m
Temperatura da parede externa (Zona de calcinação)	117.725	°C
Temperatura da parede externa (Zona de pré-aquecimento)	89.22	°C
Caudal de Combustível	0.012	kg/s
Perda de carga	927.875	Pa

Figura 6.3 - Interface gráfica de saída de valores.

O algoritmo de cálculo, o qual apenas o programador tem acesso, isto é, não é visível na interface nem mesmo durante a execução do programa, encontra-se apresentado em Anexo A.

Todos os dados necessário para iniciar a simulação são guardados em variáveis e variáveis constantes, que posteriormente ligadas entre si estabelecem as várias equações, fornecendo assim os resultados finais. Esses resultados são transportados para a interface de saída para que o utilizador possa ter acesso e percepção dos valores e resultado associados a todo o processo de cálculo.

Por fim, é também possível a opção de guardar todos os dados referentes às várias etapas de cálculo.

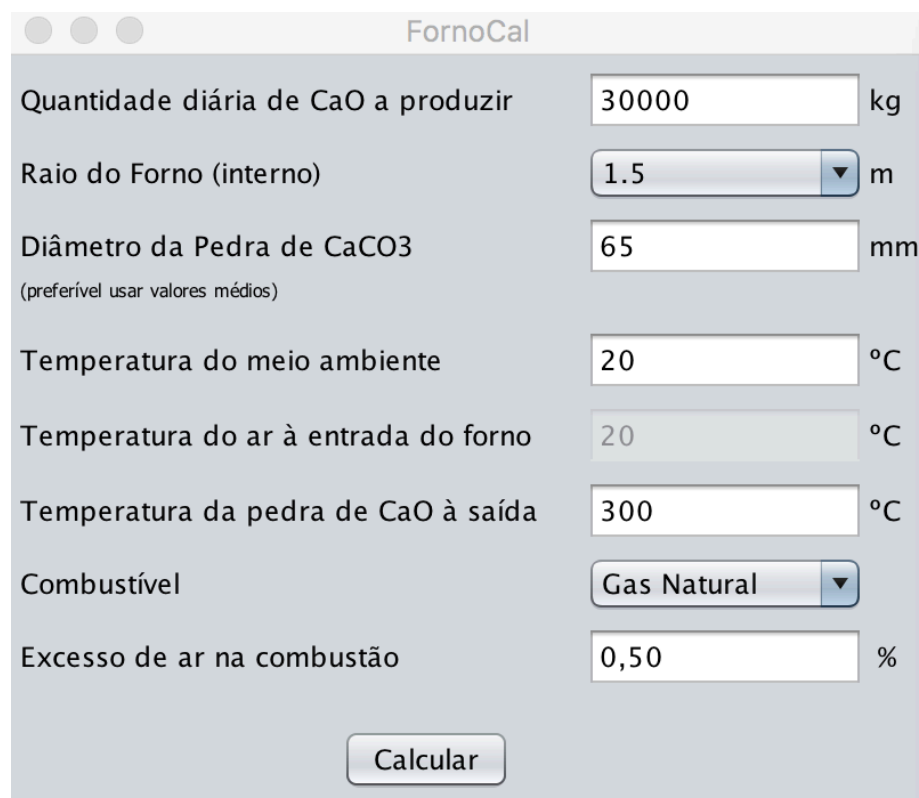
6.4 – Funcionamento da aplicação

O primeiro contacto entre o utilizador e a aplicação é a interface de entrada de valores. Nesta o utilizador necessita colocar os dados gerais, como quantidade diária de cal que pretende produzir, o raio interior do forno, a temperatura ambiente, o tipo de combustível e as condições de combustão. Todos os dados a serem impostos pelo utilizador são restringidos a uma gama de valores, a fim de evitar erros de cálculo e resultados finais anómalos.

Por último, quando a execução da aplicação for finalizada a interface de saída de resultados é, automaticamente, apresenta.

6.4.1 – Entrada de dados

A etapa de entrada de valores consiste.



The screenshot shows a software window titled "FornoCal". It contains several input fields for user data entry:

- Quantidade diária de CaO a produzir:** A text box with the value "30000" and a unit label "kg".
- Raio do Forno (interno):** A dropdown menu showing "1.5" and a unit label "m".
- Diâmetro da Pedra de CaCO₃:** A text box with the value "65" and a unit label "mm". Below this field is a note: "(preferível usar valores médios)".
- Temperatura do meio ambiente:** A text box with the value "20" and a unit label "°C".
- Temperatura do ar à entrada do forno:** A text box with the value "20" and a unit label "°C".
- Temperatura da pedra de CaO à saída:** A text box with the value "300" and a unit label "°C".
- Combustível:** A dropdown menu showing "Gas Natural".
- Excesso de ar na combustão:** A text box with the value "0,50" and a unit label "%".

At the bottom center of the window is a button labeled "Calcular".

Figura 6.4 - Interface de entrada de dados.

- Produção diária de CaO [kg]: neste campo o utilizador deve inserir a quantidade de cal que pretende produzir, com a restrição de valores entre 10000 kg a 30000 kg;
- Raio de forno [m]: pode ser escolhido um raio interno para o forno, neste campo o utilizador pode optar por um raio de 1 m ou 1,5 m;

- Diâmetro da pedra de CaCO_3 : o diâmetro da pedra à entrada pode ser indicado pelo utilizador. Este campo encontra-se restringido a valores entre 30 mm e 100 mm, no entanto o utilizador é aconselhado a optar por valores médios;
- Temperatura do meio ambiente [$^{\circ}\text{C}$]: a temperatura é considerada constante durante as 24h de operação do forno. É igualmente a temperatura de entrada do ar no forno;
- Temperatura da pedra CaO à saída [$^{\circ}\text{C}$]: a temperatura das pedras de cal à saída da zona de arrefecimento pode ser escolhida, dentro de uma gama de valores de referência;
- Combustível: o utilizador pode optar por qual o tipo de combustível pretende para a queimador. No entanto, neste programa apenas é considerado o Gás Natural devido à falta de tempo para se desenvolver um algoritmo para diversos tipos de combustível;
- Excesso de ar na combustão: as condições de combustão podem ser escolhidos pelo utilizador, a percentagem de excesso de ar na combustão é restringida a valores na gama de 20% a 50%.

6.4.2 – Desenvolvimento

Além dos valores fornecidos pelo utilizador, outros valores são utilizados no procedimento de cálculo. As propriedades físicas dos fluidos são obtidos através de polinómios associados aos mesmos. Com o conhecimento da temperatura média do filme, é possível calcular o calor específico, condutividade térmica, viscosidade cinemática, coeficiente de expansão volumétrica e o número de Prandtl. Os polinómios para a obtenção de cada propriedade foram desenvolvidos a partir dos valores dessas propriedades físicas apresentados nas tabelas de Incropera *et al.* (2013).

As propriedades físicas dos materiais refratários e isolantes, são considerados constantes no programa e foram retiradas das especificações técnicas dos produtos da empresa portuguesa Abrigada, Companhia Nacional de Refratários S.A..

O programa encontra-se dividido em 10 etapas de cálculo. Cada etapa é precedente à outra, havendo mesmo restrições no caso de erros ou valores pouco coerentes, altura em que uma mensagem é presente ao utilizador.

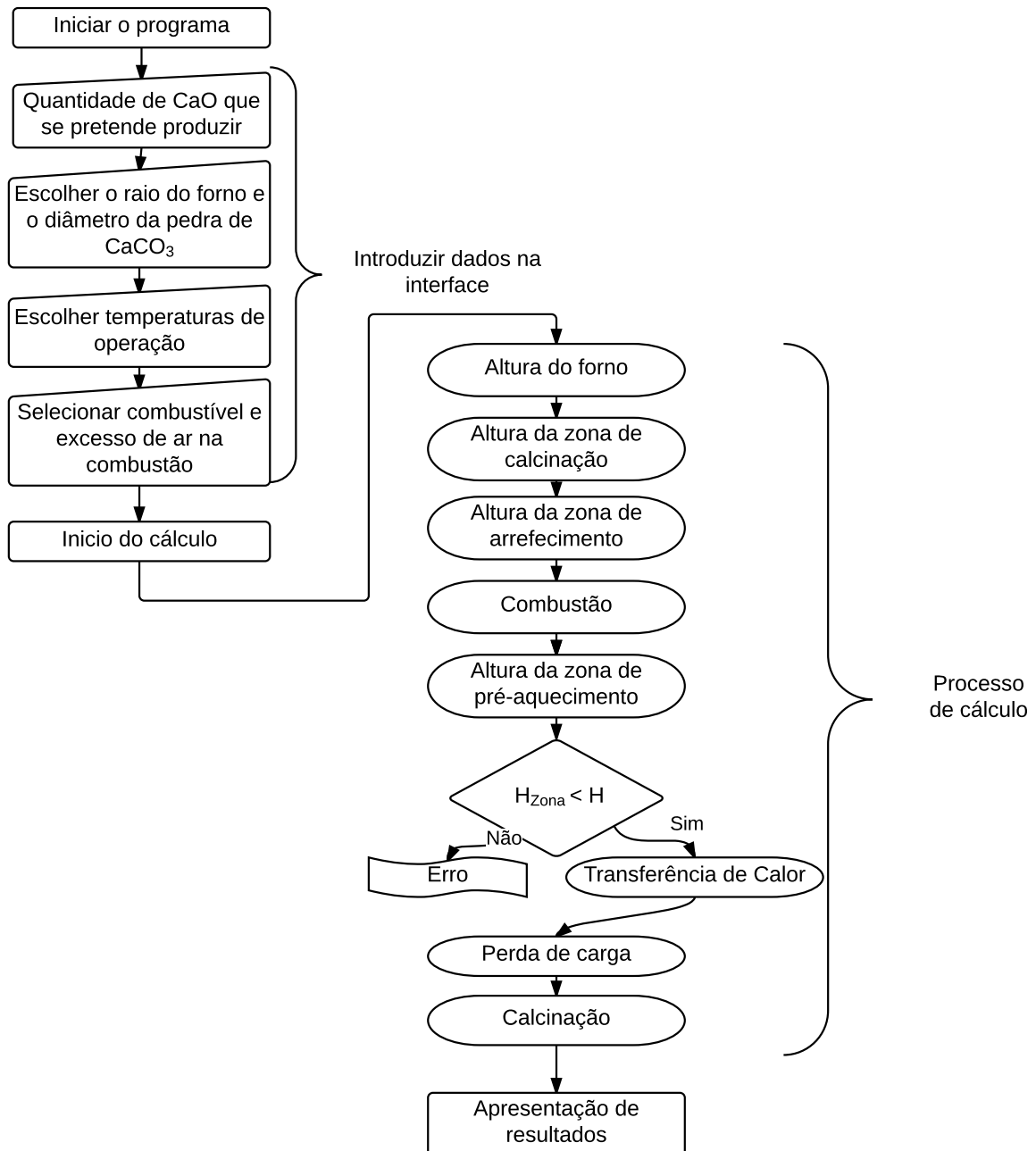


Figura 6.5 - Diagrama de funcionamento da aplicação.

6.4.3 – Saída de dados

A interface de saída consiste na apresentação dos resultados mais importantes para a construção de um forno, obtidos do procedimento de cálculo do programa.

The screenshot shows a software window titled 'FornoCal - Resultados'. It contains a list of parameters with their calculated values and units. At the bottom, there is a button labeled 'Imprimir'.

Parâmetro	Valor	Unidade
Altura do Forno	16.968	m
Zona de arrefecimento	2.44	m
Zona de calcinação	7.07	m
Zona de pré-aquecimento	1.655	m
Diâmetro do Forno (externo)	4.6	m
Espessura da parede	0.8	m
Temperatura da parede externa (Zona de calcinação)	139.355	°C
Temperatura da parede externa (Zona de pré-aquecimento)	106.319	°C
Caudal de Combustível	0.035	kg/s
Perda de carga	1505.733	Pa

Imprimir

Figura 6.6 - Interface de resultados.

- Altura do forno [m]: representa a altura total do forno.
 - Zona de arrefecimento [m]: representa a altura mínima da zona de arrefecimento;
 - Zona de calcinação [m]: representa a altura mínima da zona de calcinação;
 - Zona de pré-aquecimento [m]: representa a altura mínima da zona de pré-aquecimento, isto é apenas a altura necessário para pré-aquecer as pedras de CaCO_3 não considerando o armazenamento necessário para produzir a quantidade de CaO requerida;
- Diâmetro do forno [m]: representa o diâmetro necessário para a transferência de calor;

- Temperatura da parede exterior [°C]: esta temperatura representa apenas a temperatura na zona mais crítica, ou seja, a temperatura da parede exterior na altura da zona de calcinação;
- Caudal de combustível [kg/s]: o caudal de combustível a consumir para suportar todo o processo de dissociação da quantidade de CaCO_3 ;
- Perda de carga [Pa]: é obtido através da soma das perdas de carga associadas a cada zona.

É apresentado um botão “Imprimir” com ligação a uma janela de disposição de resultados. Esta janela trata-se um ficheiro de texto (“bloconotas.txt”) que pode ser guardo, para o caso de o utilizador pretender imprimir uma versão em papel de todo o dimensionamento. Note-se que o separador decimal dos resultados é representado por o ponto decimal.

```

Fase 1: Altura do forno -----
0 volume da partícula (Vp) é [m3]: 0.001437933137486828
0 volume liquido total (VliqT) é [m3]: 19.841269841269842
0 número de pedras por unidade de volume (Np) é: 4203.950686166987
0 número de pedras de CaCO3 no interior do forno (nPt) é: 83411.71996363069
0 volume do forno (Vt) é [m3]: 119.94047619047618
A altura total do forno (H) é [m]: 16.968106366675613
----- Fim da Fase 1 -----

----- Fase 2: Altura da zona de calcinação -----
A quantidade de Massa de CaCO3 na calcinação é [kg]: 22321.42857142857
0 volume liquido na zona de calcinação (VliqC) é [m3]: 8.267195767195766
0 número de pedras na zona de calcinação (nPc) é: 34754.88331817945
0 volume da zona de calcinação (Vtc) é [m3]: 49.9751984126984
A altura da zona de calcinação (H2) é [m]: 7.07004431944817
----- Fim da Fase 2 -----

----- Fase 3: Altura da zona de arrefecimento -----
A área de transferência de calor (At) é [m2]: 0.0033183072403542195
0 número de Reynolds (Re1) é: 1025.6586169045006
0 caudal de ar (mG1) é [kg/s]: 3.657991946023615
0 coeficiente de transferência de calor por convecção (h1) é [W/m2K]: 0.05066372409836624
A quantidade de energia (Qg1) é [kW]: 3.0554880432253673
A altura da zona de arrefecimento é (H1): 2.439671156829984
----- Fim da Fase 3 -----

----- Fase 4: Combustão -----
Fração molar do CO2 é: 0.10009640545379424
Fração molar do H2O é: 0.18634715144837719
Fração molar do N2 é: 0.6168434099986229
Fração molar do He é: 1.8362943579855852E-4
Fração molar do O2 é: 0.09652940366340722
Fração mássica do CO2 é: 0.1561106535892871
Fração mássica do H2O é: 0.11889309896113118
Fração mássica do N2 é: 0.6154810870523273
Fração mássica do He é: 2.603536525229518E-5
Fração mássica do O2 é: 0.10948912503200212
0 calor específico dos produtos (Cp) é: 1.411884645766197
0 calor necessário para a reação é: 1.3452E8
0 PCI é: 45109.4716801523
0 caudal de Combustível por dia (mCombDia) é [kg/dia]: 2982.0788182537594
0 caudal de Combustível necessário (mComb) é [kg/s]: 0.03451480113719629
A massa dos produtos (mp) é [kg]: 307.2743524999999
Razão Produtos/Combustível é: 16.396710378868725
0 caudal dos produtos (dotMp) é [kg/s]: 0.5659291980308564
----- Fim da Fase 4 -----

----- Fase 5: Altura da zona de pré-aquecimento -----
0 número de Reynolds (Re3) é: 733.2554116829308
0 caudal de gases (mG3) é [kg/s]: 3.2310495044007626
0 caudal de dióxido de carbono (mCO2) é [kg/s]: 0.27281746031746035
0 caudal total de gases (m3) é [kg/s]: 4.069796162749079
0 coeficiente de transferência de calor por convecção (h3) é [W/m2K]:

```

Figura 6.7 - Ficheiro texto dos resultados.

Capítulo 7

Resultados da simulação

Neste capítulo serão apresentados os resultados obtidos de uma simulação. Onde são comparados com resultados obtidos no procedimento de cálculo manual, realizado anteriormente com recurso à ferramenta Microsoft Office Excel.

7.1 – Simulação

É importante salientar que, devido a constantes e imprevisíveis mudanças nas condições ambientes e de trabalho dos fornos, os resultados das simulações não devem ser usados diretamente no controle do processo mas sim interpretados com cautela. Neste contexto, é apresentada uma avaliação do programa de simulação computacional desenvolvido neste trabalho e a análise dos principais parâmetros de operação de um forno vertical. Inicialmente, são apresentados os resultados de uma simulação realizada com valores típicos médios de entrada. Em seguida, esses resultados são comparados com valores resultantes de uma análise anterior realizada no Excel. Por fim, é realizado uma análise dos aspetos que influenciam os principais parâmetros de operação do forno, baseados em resultados de simulações obtidos com a utilização do programa de simulação.

Diversas simulações foram realizadas, visando o estudo de um forno vertical operando sob as mesmas condições, sendo estas, a produção diária, temperatura de ar ambiente e tipo de combustível. Os principais valores de entrada destas simulações são apresentados na Tabela 7.1.

Tabela 7.1 - Dados de entrada.

	Valor	Unidade
Quantidade diária de CaO a produzir	30000	kg
Diâmetro da pedra de CaCO_3 (d_p)	65	mm
Raio interno do forno (R)	1,5	m
Temperatura do ar ambiente	20	°C
Temperatura da pedra de CaO à saída do forno	300	°C
Excesso de ar	50	%

Esta simulação é realizada para a capacidade máxima da carga do forno (30 t), com um diâmetro médio da pedra de CaCO_3 e com um excesso de ar na combustão máximo (50%).

Tabela 7.2 – Resultados obtidos da simulação.

	Valor	Unidade
Altura do forno	16,97	m
Altura zona de arrefecimento	2,44	m
Altura zona de calcinação	7,07	m
Altura zona de pré-aquecimento	1,65	m
Diâmetro externo do forno	4,6	m
Espessura das paredes	0,80	m
Caudal de combustível	0,035	kg/s
Caudal de combustível diário	2982,08	kg/dia
Temperatura da parede externa na zona crítica (zona calcinação)	139,35	°C
Temperatura da parede externa na zona de passagem dos gases de escape (zona pré-aquecimento)	106,32	°C
Perda de carga	1505,73	Pa

A Tabela 7.3 apresenta-se uma comparação entre os resultados obtidos no programa de simulação e os do cálculo realizado inicialmente com Microsoft Office Excel. Esta comparação consiste numa avaliação do dimensionamento do forno, e comportamento térmico.

Tabela 7.3 - Comparação dos resultados obtidos.

	Programa de simulação	Excel	Erro
Altura forno [m]	16,97	16,968	0,01%
Altura da zona de arrefecimento [m]	2,44	2,44	-
Altura da zona de calcinação [m]	7,07	7,07	-
Altura da zona de pré-aquecimento [m]	1,65	1,748	5,34%
Diâmetro externo do forno [m]	4,6	4,6	-
Espessura das paredes [m]	0,8	0,8	-
Caudal combustível [kg/s]	0,0345	0,0345	-
Temperatura da parede externa na zona crítica (zona calcinação) [°C]	139,35	139,56	0,15%
Temperatura da parede externa na zona de passagem dos gases de escape (zona pré-aquecimento) [°C]	106,32	108,51	2,02%
Perda de carga [Pa]	1505,73	1521	1,00%

Embora os resultados dos dois cálculos apresentem valores relativamente próximos, é possível verificar que certos valores apresentam erros relativos maiores em relações a outros. A razão para tal reside da forma como são calculadas as propriedades físicas do fluido (ar e gases de combustão), uma vez que, no programa de simulação tais valores foram obtidos através de polinómios, enquanto que no cálculo via Excel foram simplesmente retirados das tabelas termodinâmicas existentes em Incropera *et al.* (2013).

Tabela 7.4 - Comparação de resultados em relação ao excesso de ar na combustão.

Excesso de ar		Altura da zona de Pré-aquecimento [m]	Erro	Caudal de gases na zona de pré-aquecimento [kg/s]	Erro
20%	Prog.	1,629	1,48%	4,178	0,024%
	Excel	1,653		4,179	
40%	Prog.	1,645	4,22%	4,285	0,014%
	Excel	1,718		4,286	
50%	Prog.	1,655	5,34%	4,34	0,025%
	Excel	1,748		4,339	

O excesso de ar de combustão, parâmetro essencial para garantir a queima limpa e completa do combustível, afeta o caudal de gases no interior do forno. Uma vez que as propriedades físicas dos gases, viscosidade dinâmica, condutibilidade térmica e calor específico, dependem da combustão e esta por sua vez é influenciada pelo excesso de ar de combustão é de esperar que as propriedades físicas sofram alterações com diferentes valores de excesso de ar. A condutibilidade térmica ao diminuir com o aumento do excesso de ar faz com que o coeficiente de transferência de calor por convecção diminua, reduzindo-se a taxa de transferência de calor que, consequentemente, aumenta o comprimento do forno necessário de pré-aquecimento que as partículas devem percorrer para atingir a temperatura desejada (Tabela 7.4).

Embora havendo um maior excesso de ar, como este leva a um abaixamento da temperatura da corrente gasosa, tal deveria levar a uma redução da viscosidade. Contudo como há alterações na composição da corrente gasosa verifica-se um ligeiro aumento da viscosidade. Esta subida compreende-se porque a viscosidade do ar é superior à do CO_2 e do H_2O . Logo, mais excesso de ar significa um aumento da viscosidade. Por outro lado, nesta análise do efeito de excesso de ar, assumiu-se sempre a mesma temperatura dos gases, no entanto tal não é verdade pois na realidade, ao subir o excesso de ar, baixa a temperatura da chama. Embora a temperatura da chama desça com o aumento do excesso de ar, tal não se verifica, tratando-se isto de uma limitação do programa, pois as condições de combustão não foram tidas em conta nesta análise.

Em resumo, com o aumento do excesso de ar de combustão obtém-se um menor coeficiente de transferência de calor, uma maior perda de carga devida tanto à maior viscosidade como ao maior caudal envolvido e um menor diferencial térmico entre a referida corrente gasosa e as pedras a tratar.

Tabela 7.5 - Valores das propriedades físicas dos gases para diferentes valores de excesso de ar.

Excesso de ar	20%	40%	50%	
Viscosidade dinâmica (μ)	3,99E-05	4,03E-05	4,05E-05	Pa s
Condutibilidade térmica (k_t)	0,1121	0,1097	0,1087	W/m K

Capítulo 8

Conclusões e sugestões

Este capítulo dedica-se à apresentação das principais conclusões obtidas neste estudo, além de fazer referência a algumas sugestões para trabalhos futuros.

8.1. – Conclusão

A construção de um programa de dimensionamento de fornos de cal verticais consistiu na elaboração de um algoritmo que permite a automatização de todo o processo de dimensionamento deste sistema

Os dados de entrada do programa de simulação desenvolvido neste trabalho abrangem os principais parâmetros de projeto e operação do forno, referentes à sua geometria (raio interno, isolamento térmico), produção (produção diária, tempo de residência), e consumo de combustível (consumo de gás natural e excesso de ar de combustão).

O desempenho e a aplicabilidade prática desta aplicação, pode ser para já, considerado bastante positivo, embora careça de um período exaustivo de testes por utilizadores experientes nesta área.

- O calcário, por se tratar de uma matéria-prima mineral, apresenta grande variabilidade de composição e estrutura morfológica em geral. Por essa razão, a caracterização deste torna-se de uma etapa imprescindível aos estudos de otimização das condições de operação e mesmo do acompanhamento da qualidade e homogeneidade da cal produzida. Para se solucionar esse impasse, é fundamental que se proceda com uma

etapa inicial de mapeamento da mina da qual se extrai o calcário em termos de composição e morfologia.

- A presença de um caudal mínimo de gás de arrasto é de fundamental importância para que haja uma efetiva calcinação do calcário. A necessidade de inserção de gás (ar) no interior dos fornos industriais tem como objetivo, além de fornecer comburente à queima do combustível, o retirar o dióxido de carbono liberado durante o processo reacional, de forma a reduzir as limitações termodinâmicas.
- O tamanho das partículas de calcário desempenha um papel importante na operação do forno. Um diâmetro da partícula adequado deve ser usado no forno. Pequenos pedaços de pedra aumentam a resistência ao escoamento de gás no interior do forno o que pode provocar uma redução na taxa de produção.
- As partículas carregadas nos fornos de cal são irregulares e não perfeitamente esféricas, o que é diferente do que pressuposto no modelo de cálculo deste forno, de modo que os resultados das simulações devem ser interpretados com cautela.
- A geometria do forno definida deste modelo favorece o escoamento dos gases de combustão, garantindo assim uma maior intensidade dos fenômenos de transferência de calor no interior do forno.

Finalmente, espera-se com este trabalho ampliar e difundir o conhecimento sobre o funcionamento de fornos de cal verticais, de forma a auxiliar no projeto de novos fornos e no ajuste daqueles já em operação.

8.2. – Trabalhos Futuros

A realização deste trabalho suscitou diversas questões acerca da operação e do comportamento térmico dos fornos. Assim, podem ser sugeridas diversas possibilidades para estudos futuros com a utilização do programa de simulação desenvolvido:

- Construção de uma instalação piloto para verificação de outros efeitos, tais como tipo de combustível na qualidade da cal;
- Aplicação do programa no projeto de fornos calcinação visando otimizar a produção de cal em termos de produtividade e redução de custos energéticos;

- Estudo sistemático da eficiência energética de fornos verticais com diferentes tipos e espessuras de paredes refratárias e isolantes, a fim de fornecer subsídios para a determinação da relação ótima entre os custos de construção e de operação do forno.

Além disso, como o programa de simulação foi desenvolvido desde o início durante este trabalho vislumbrando o seu constante aprimoramento, algumas melhorias podem ainda ser implementadas no código computacional. Ficam como principais sugestões para o desenvolvimento futuro do programa de simulação, as seguintes observações:

- Modelação do processo produtivo em programas de dinâmica de fluidos computacional;
- Estabelecer uma base de dados com a metodologia de produção de cales de diferentes especificidades, de acordo com o requerimento de sectores consumidores dessa matéria-prima;
- Aprimoramento do modelo de cálculo do tempo de residência da carga no interior do forno;
- Inserção de equações do balanço de energia de transferência de calor por radiação entre os volumes de controlo, representando também os gases de combustão, as paredes e a carga, a fim de se aproximar o modelo matemática da realidade;
- Inserção de equações dos balanços de energia e de massa, de geração ou absorção de energia, devido às reações físicas e químicas que podem ocorrer na carga e nos gases do forno;

Uma vez que a indústria de cal está entre as mais poluidoras do meio ambiente. Seria importante tomar em consideração combustíveis ou alternativas à combustão para reduzir a emissão de CO₂ produzido pela queima do combustível.

Num aspeto mais global e a fim de facilitar o projeto de qualquer tipo de forno de calcinação, seria interessante realizar um modelo matemático para todo o tipo e características de fornos, desde a vasta gama de fornos verticais até aos fornos rotativos (entre outros).

Referências

- Anderson, (1973); *Resumé of ICI Work on Limestone Calcination, Lime Reactivity and Apparent Density*.
- Azbe, V. J. (1953); *Theory and Practice of Lime Manufacture*, Part II, Rock Products, p 102-104.
- Bauer, L. A. F. (1987); *Materiais de Construção 1*. Livros Técnicos e Científicos Editora, 3ª Edição, Rio de Janeiro – Brasil.
- Beach, D. H. (1983); *Chemical of the month*, Vol 60, p 60 – 63.
- Boynton, R.S. (1980); *Chemistry and Technology of lime and limestone*. John Wiley & Sons, New York, U.S.A.
- Ciaccia, P. M. (1974); *Aperfeiçoamentos em fornos rotativos de calcinação e terminologia*. Metalurgia, São Paulo, Brasil.
- European Commission, (2013); *Reference Documente on Best Available Techniques in the Cement and Lime Manufacturing Industries*, Abril 2013, Bruxelas.
- Garcia-Labiano, F., Abad, A., Diego, L. F., Gayan, P. e Adanez, J. (2002); *Calcination of calcium-based sorbents at pressure in a broad range of CO₂ concentrations*. Chemical Engineering Science, Vol. 57, p. 2381–2393,
- Gamej, A.I., Perlevskij, A., Kurchenko O., Bol'shakova, Z. e Malikova, E. (2004); *Improvement in quality of preparation of limestone charged into the firing units*. Stal, Vol. 12, p. 9 – 11.
- Guimarães, J. E. P. (2002); *A cal – fundamentos e aplicações na engenharia civil*. 2ª Edição, São Paulo.

- Jauhari, R., Gray, M.R. e Masliya, J.H. (1998); *Gas-solid mass transfer in a rotating drum*. Canadian Journal of Chemical Engineering, N. 76, p. 224 – 232.
- Incropera, F.P., Dewitt, D.P., Bergman, T.L. e Lavine, A.S. (2013), *Fundamentals of Heat and Mass Transfer*, 7th Edition, John Wiley & Sons, New York.
- Kawasaki Heavy Industries (1983); *Explanations on Kawasaki's lime calcining plant* – Dedini Kawasaki Engenharia S.A., São Paulo.
- Kinsler, T. J. (1991); *Operation of calcimatic kilns*, National Lime Association, Meeting 1991, Arlington, U.S.A.
- Kramers, H. e Crookewit, P. (1952); *The passage of granular solid through inclined rotary kilns*. Chemical Engineering Science, N. 1, p. 259.
- Koike, K. e Matsuda, S. (2003); *Characterizing content distributions of impurities in a limestone mine using a feed forward neural network*. Natural Resources Research, Vol. 12, N. 3.
- Lebas, E.F., Hanrot, F., Ablitzer, D. e Houzelot, J.L. (1995); *Experimental study of residence time, particle movement and bed depth profile in rotary kilns*. Canadian Journal of Chemical Engineering, N. 73, p. 173-179.
- Leslie, A. B. e Hughes, J. J. (2004); *High-temperature slag formation in historic Scottish mortars: evidence for production dynamics in 18th 19th century lime production from Charlestown*. Materials Characterization, Vol. 53, p. 181-186.
- Levenspiel, O (1998); *Engineering flow and heat exchange*. Rev. Ed., Plenum Press, New York.
- LNEG, Laboratório Nacional de Energia e Geologia, <http://www.lneg.pt>, acessado em 15 de Outubro 2015.
- Manfred, D. R. e Grinfeder, C. A. (1976); *Fornos Rotativos, Forno Lepol para a fabricação de cal virgem* – 1º Seminário sobre operação de fornos, Associação Brasileira dos Produtores de Cal, São Paulo, Brasil.
- Meier, A., Bonaldi, E., Cella, G.M., Lipinski, W., Palumbo, R., Steinfeld, A., Wieckert, C. E Wuillemmin, D. (2004); *Towards the industrial production of lime*, PSI Scientific Report 2003, Volume 5, p. 31 – 32.

- Mereu, F. (1994); *Maerz Vertical Limes Shaft Kiln for the Calcination of Small Size Limestone*, Proc. 8th International Lime Congress, Berlin.
- Millet, J. e Alii (1979); *Influência da temperatura de calcinação e da natureza do calcário sobre a mineralogia das cales virgens*, tradução Maria Cincotto, Associação Brasileira dos Produtos de cal. N.T. 71-1979, São Paulo, Brasil,
- National Lime Association (1957); *Chemical Lime Facts*, Bul. 214, Arlington, U.S.A.
- Oates, J.A.H. (1998); *Lime and Limestone: Chemistry and Technology, Production and Uses*. Wiley-VCH, Weinheim.
- Ochoa, P.A.G. et al. (2010); *Cleaner production in a small lime factory by means of process control*, Journal of Cleaner Production, Vol.18, N.12, p. 1171 – 1176.
- Paiva, S.C.; Gomes, E.A.; Oliveira, R.A. (2007); *Controle de Qualidade da Cal para Argamassas – Metodologias Alternativas*. Revista Ciência & Tecnologia. Ano 1, Nº 1, p.1 – 11.
- Perona, F. (1976); *Fornos verticais Azbe: Operação*, 1º Seminário sobre operação de fornos, Associação Brasileira dos Produtos de Cal, São Paulo, Brasil.
- Pinho, C., (2011), *Combustão, Anexo II – Física da Combustão*. Departamento de Engenharia Mecânica, FEUP, p. 296 – 297, Porto, Portugal.
- Predescu, L. (1989); *Pleading for the vertical lime kiln*, Revue Générale de Thermique, p. 273 – 279.
- Quallical, www.quallical.com.br, acedido em 4 em Novembro de 2015.
- Rovaglio, M., Manca, D., Biardi, G. (1998); *Dynamic Modeling of Waste Incineration Plants With Rotary Kilns*, Chemical Engineering Science, Vol. 53, N 15, p. 2727 – 2742.
- Ruckensteiner, G., Burczech, J. e Ludwing, U. (1995); *Factors Affecting the Thermal Stability of the Internal Structure When Burning Limestone*; Zement Kalk Gyps 6.
- Saeman, W. C. (1951); *Passage of solids through rotary kilns*; Chemical Engineering Progress, N. 47, p. 508-514.

- Schwarzkorf, F (1978); *Lime burning technology: A manual for lime plant operators*, Kennedy Van Saun Corporation, Danville, U.S.A.
- Scott, G. D. (1960); *Packing of spheres Nature*, N. 188, p. 908 – 911.
- Shreve, R. N., e Brink, J. A. Jr. (1977); *Chemical Process Industries*, 4th Ed, p. 156 – 178.
- Silcox, G.D., Kramlich, J.C. e Pershing, D.W. (1989); *A mathematical model for the flash calcination of dispersed CaCO_3 and Ca(OH)_2 particles*, Ind. Eng. Chem. Res., Vol. 28, p. 155 – 160.
- Smidth, F.L.S. (1975); *Lime kiln systems*, F.L.S., Folheto 1975, Copenhagen, Dinamarca.
- Spurling, R.J., Davidson, J.F. e Scott, D.M. (2000); *The no-flow problem for granular material in rotating kilns and dish granulators*; Chemical Engineering Science, N. 55, p. 2303 – 2313.
- Sullivan, J.D., Maier, C.G. e Ralston, O.C. (1927); *Passage of solid particles through rotary cylindrical kilns*; *US Bureaus of Mines*, Technical Papers 384, p. 1 – 42.
- Thompson, J. L. (1979); *“Prediction Lime Burning Rate via New Dynamic Calcination Theory”*, Pit and Quarry 5.
- Tran, H.N., Vollick, S., Gauthier, M. e McNeil, C., (2005); *Correlation between nodule size and residual carbonate content in lime kiln*; Proceedings of Tappi Environmental Pulping and Engineering Conference, Philadelphia.
- Wikipédia, <https://pt.wikipedia.org/wiki/Pré-Cambriano>, acedido em 15 de Outubro de 2015.
- Yang, W-C. (2003); *Handbook of Fluidization and Fluid-Particle Systems*, Marcel Dekker, Inc., New York, U.S.A.
- Zou, R.P. e Yu, A.B., (1995), *The parking of spheres in a cylindrical container: the thickness effect*, Chem. Eng. Sci, p. 1504 – 1507.

Anexo A

Algoritmo do Programa

```
package Fase1;

public class Constants {
    public final static int RhoCaCO3 = 2700;
    public final static double E = 0.3955;
    public final static double PI = Math.PI;
}

public class Variables {
    public static int quantidadeDeCaO;
    public static double quantidadeDeCaCO3;
    public static double dp;
    public static double R;
    public static double Vp;
    public static double VliqT;
    public static double Np;
    public static double nPt;
    public static double Vt;
    public static double H;

    public void definirVariaveisUtilizador(int quantCaO_user, double quantCaCO3_user, double
dp_user, double R_user){
        quantidadeDeCaO = quantCaO_user;
        quantidadeDeCaCO3 = quantCaCO3_user;
        dp = dp_user;
        R = R_user;
    }

    public void definirVariaveis(double Vp_user, double VliqT_user, double Np_user, double nPt_user,
double Vt_user, double H_user){
        Vp = Vp_user;
        VliqT = VliqT_user;
        Np = Np_user;
        nPt = nPt_user;
        Vt = Vt_user;
        H = H_user;
    }

    public int getQuantidadeDeCaO() {
        return quantidadeDeCaO;
    }

    public double getQuantidadeDeCaCO3() {
        return quantidadeDeCaCO3;
    }
}
```

```
}  
public double getDp() {  
    return dp;  
}  
public double getR() {  
    return R;  
}  
public double getVp() {  
    return Vp;  
}  
public double getVliqT() {  
    return VliqT;  
}  
public double getNp() {  
    return Np;  
}  
public double getnPt() {  
    return nPt;  
}  
public double getVt() {  
    return Vt;  
}  
public double getH() {  
    return H;  
}  
}  
package Fase1;  
import Viewer.Resultados;  
public class Equations {  
public static void equation1(){  
    Resultados.resultados.append("----- Fase 1: Altura do forno -----  
").append(System.getProperty("line.separator"));  
    Variables.Vp = equation1_Vp(Variables.dp);  
    System.out.println("A Quantidade de Vp é agora de: " + Variables.Vp);  
    Resultados.resultados.append("O volume da partícula (Vp) é [m3]:  
").append(Variables.Vp).append(System.getProperty("line.separator"));  
    Variables.VliqT = equation1_vLigT(Variables.quantidadeDeCaCO3);  
    System.out.println("A Quantidade de VliqT é agora de: " + Variables.VliqT);
```

```

        Resultados.resultados.append("O volume liquido total (VliqT) é [m3]:
").append(Variables.VliqT).append(System.getProperty("line.separator"));
        Variables.Np = equation1_Np(Variables.dp);
        System.out.println("A Quantidade de Np é agora de: " + Variables.Np);
        Resultados.resultados.append("O número de pedras por unidade de volume (Np) é:
").append(Variables.Np).append(System.getProperty("line.separator"));
        Variables.nPt = equation1_npt(Variables.Np, Variables.VliqT);
        System.out.println("A Quantidade de nPt é agora de: " + Variables.nPt);
        Resultados.resultados.append("O número de pedras de CaCO3 no interior do forno (nPt)
é:").append(Variables.nPt).append(System.getProperty("line.separator"));
        Variables.Vt = equation1_vt(Variables.nPt, Variables.Vp);
        System.out.println("A Quantidade de Vt é agora de: " + Variables.Vt);
        Resultados.resultados.append("O volume do forno (Vt) é [m3]:
").append(Variables.Vt).append(System.getProperty("line.separator"));
        Variables.H = equation1_H(Variables.Vt, Variables.R);
        System.out.println("A altura total do forno é (H): " + Variables.H);
        Resultados.resultados.append("A altura total do forno (H) é [m]:
").append(Variables.H).append(System.getProperty("line.separator"));
        Resultados.resultados.append("----- Fim da Fase 1 -----
").append(System.getProperty("line.separator")).append(System.getProperty("line.separator"));
    }

    public static double equation1_Vp(double dp){
        double vp = 0.0;
        vp = (((4*Math.PI)/3)*(Math.pow(((dp*0.001)/2),3))) * 10; //m3
        return vp;
    }

    public static double equation1_vLigT(double quantCaCO3){
        double vLigT = 0.0;
        vLigT = quantCaCO3 / Constants.RhoCaCO3; //m3
        return vLigT;
    }

    public static double equation1_Np(double dp){
        double Np = 0.0;
        Np = (6*(1-Constants.E))/(Math.PI*(Math.pow((dp*0.001),3)));
        return Np;
    }

    public static double equation1_npt(double Np, double vLigT){
        double npt = 0.0;

```



```
npt = Np * vLigT;
return npt;
}
public static double equation1_vt(double npt, double vp){
    double vt = 0.0;
    vt = npt * vp;
    return vt;
}
public static double equation1_H(double vt, double R){
    double H = 0.0;
    H = vt/(Math.PI*Math.pow(R,2)); //m
    return H;
}
}
package Fase2;
public class Variables {
    public static double MassaCaCO3calcinacao;
    public static double VliqC;
    public static double nPc;
    public static double VtC;
    public static double H2;
    public void definirVariaveis(double MassaCaCO3c_user, double VliqC_user, double nPc_user,
double VtC_user, double H2_user){
        MassaCaCO3calcinacao = MassaCaCO3c_user;
        VliqC = VliqC_user;
        nPc = nPc_user;
        VtC = VtC_user;
        H2 = H2_user;
    }
    public double getMassaCaCO3calcinacao() {
        return MassaCaCO3calcinacao;
    }
    public double getVliqC() {
        return VliqC;
    }
    public double getnPc() {
        return nPc;
    }
}
```

```

public double getVtC() {
    return VtC;
}

public double getH2() {
    return H2;
}
}

package Fase2;
import Fase1.Constants;
import Viewer.Resultados;
public class Equations {
    public static void equation2() {
        Resultados.resultados.append("----- Fase 2: Altura da zona de calcinação -----");
        ").append(System.getProperty("line.separator"));
        Variables.MassaCaCO3calcinacao =
equation2_MassaCaCO3calcinacao(Fase1.Variables.quantidadeDeCaO);
        System.out.println("A quantidade de Massa de CaCO3 na calcinação é de: " +
Variables.MassaCaCO3calcinacao);
        Resultados.resultados.append("A quantidade de Massa de CaCO3 na calcinação é [kg]:");
        ").append(Variables.MassaCaCO3calcinacao).append(System.getProperty("line.separator"));
        Variables.VliqC = equation2_VliqC(Variables.MassaCaCO3calcinacao);
        System.out.println("A quantidade de VliqC é de: " + Variables.VliqC);
        Resultados.resultados.append("O volume liquido na zona de calcinação (VliqC) é [m3]:");
        ").append(Variables.VliqC).append(System.getProperty("line.separator"));
        Variables.nPc = equation2_nPc(Variables.VliqC, Fase1.Variables.Np);
        System.out.println("A quantidade de nPc é de: " + Variables.nPc);
        Resultados.resultados.append("O número de pedras na zona de calcinação (nPc) é:");
        ").append(Variables.nPc).append(System.getProperty("line.separator"));
        Variables.VtC = equation2_VtC(Variables.nPc, Fase1.Variables.Vp);
        System.out.println("A quantidade de Vtc é de: " + Variables.VtC);
        Resultados.resultados.append("O volume da zona de calcinação (Vtc) é [m3]:");
        ").append(Variables.VtC).append(System.getProperty("line.separator"));
        Variables.H2 = equation2_H2(Variables.VtC, Fase1.Variables.R);
        System.out.println("A altura da zona de calcinação é (H2): " + Variables.H2);
        Resultados.resultados.append("A altura da zona de calcinação (H2) é [m]:");
        ").append(Variables.H2).append(System.getProperty("line.separator"));
        Resultados.resultados.append("----- Fim da Fase 2 -----");
        ").append(System.getProperty("line.separator")).append(System.getProperty("line.separator"));
    }
}

```

```
}  
public static double equation2_MassaCaCO3calcinacao(double quantCaO){  
    double MassaCaCO3calcinacao = 0.0;  
    MassaCaCO3calcinacao = (Fase1.Variables.quantidadeDeCaO*10*0.1)/(24*0.056); //kg de  
CaCO3  
    return MassaCaCO3calcinacao;  
}  
public static double equation2_VliqC(double MassaCaCO3calcinacao){  
    double VliqC = 0.0;  
    VliqC = (MassaCaCO3calcinacao/Constants.RhoCaCO3); //m3  
    return VliqC;  
}  
public static double equation2_nPc(double VliqC, double Np){  
    double nPc = 0.0;  
    nPc = Np*VliqC;  
    return nPc;  
}  
public static double equation2_VtC(double nPc, double vp){  
    double VtC = 0.0;  
    VtC = nPc*vp; //m3  
    return VtC;  
}  
public static double equation2_H2(double VtC, double R){  
    double H2 = 0.0;  
    H2 = VtC/(Math.PI*Math.pow(R, 2)); //m  
    return H2;  
}  
}  
package Fase3;  
public class Constants {  
    public final static double Uf1 = 1;  
    public final static double RhoG1 = 0.5175;  
    public final static double Cp1 = 1.070886;  
    public final static double muG1 = 0.000032796;  
    public final static double Pr1 = 0.6833;  
    public final static double kG1 = 0.000051401;  
    public final static int Tgc1 = 800;  
    public final static int Tpc1 = 850;
```

```
package Fase3;

public class Variables {

    public static int Tps1;
    public static int Tge1;
    public static double At;
    public static double Ab;
    public static double Re1;
    public static double mG1;
    public static double h1;
    public static double Qg1;
    public static double H1;

    public void definirVariaveisUtilizador(int Tps1_user, int Tge1_user){
        Tps1 = Tps1_user;
        Tge1 = Tge1_user;
    }

    public void definirVariaveis(double At_user, double Ab_user, double Re1_user, double mG1_user,
double h1_user, double Qg1_user, double H1_user){
        At = At_user;
        Ab = Ab_user;
        Re1 = Re1_user;
        mG1 = mG1_user;
        h1 = h1_user;
        Qg1 = Qg1_user;
        H1 = H1_user;
    }

    public int getTps1() {
        return Tps1;
    }

    public int getTge1() {
        return Tge1;
    }

    public double getAt() {
        return At;
    }

    public double getAb() {
        return Ab;
    }

    public double getRe1() {
```

```
        return Re1;
    }

    public double getmG1() {
        return mG1;
    }

    public double geth1() {
        return h1;
    }

    public double getQg1() {
        return Qg1;
    }

    public double getH1() {
        return H1;
    }
}

package Fase3;
import Viewer.Resultados;
public class Equations {
    public static void equation3() {
        Resultados.resultados.append("----- Fase 3: Altura da zona de arrefecimento -----");
        ").append(System.getProperty("line.separator"));
        Variables.At = equation3_At(Fase1.Variables.dp);
        System.out.println("A quantidade de At é: " + Variables.At);
        Resultados.resultados.append("A área de transferência de calor (At) é [m2:");
        ").append(Variables.At).append(System.getProperty("line.separator"));
        Variables.Ab = equation3_Ab(Fase1.Variables.R);
        System.out.println("A quantidade de Ab é: " + Variables.Ab);
        Variables.Re1 = equation3_Re1(Fase1.Variables.dp);
        System.out.println("A quantidade de Re1 é: " + Variables.Re1);
        Resultados.resultados.append("O número de Reynolds (Re1) é:");
        ").append(Variables.Re1).append(System.getProperty("line.separator"));
        Variables.mG1 = equation3_mG1(Variables.Ab);
        System.out.println("A quantidade de mG1 é: " + Variables.mG1);
        Resultados.resultados.append("O caudal de ar (mG1) é [kg/s:");
        ").append(Variables.mG1).append(System.getProperty("line.separator"));
        Variables.h1 = equation3_h1(Fase1.Variables.dp, Variables.Re1);
        System.out.println("A quantidade de h1 é: " + Variables.h1);
```

```

Resultados.resultados.append("O coeficiente de transferência de calor por convecção (h1) é
[W/m2K]: ").append(Variables.h1).append(System.getProperty("line.separator"));
Variables.Qg1 = equation3_Qg1(Variables.mG1, Variables.Tge1);
System.out.println("A quantidade de Qg1 é: " + Variables.Qg1);
Resultados.resultados.append("A quantidade de energia (Qg1) é [kW]:
").append(Variables.Qg1).append(System.getProperty("line.separator"));
Variables.H1 = equation3_H1(Variables.Qg1, Fase1.Variables.dp, Fase3.Variables.Tps1,
Fase3.Variables.Tge1, Variables.h1, Variables.At);
System.out.println("A altura da zona de arrefecimento é (H1): " + Variables.H1);
Resultados.resultados.append("A altura da zona de arrefecimento é (H1):
").append(Variables.H1).append(System.getProperty("line.separator"));
Resultados.resultados.append("----- Fim da Fase 3 -----
").append(System.getProperty("line.separator")).append(System.getProperty("line.separator"));
}
public static double equation3_At(double dp){
    double At = 0.0;
    At = ((Math.PI)*(Math.pow((dp*0.001),2)))/4; //m2
    return At;
}
public static double equation3_Ab(double R){
    double Ab = 0.0;
    Ab = (Math.PI)*(Math.pow(R,2)); //m2
    return Ab;
}
public static double equation3_Re1(double dp){
    double Re1 = 0.0;
    Re1 = (dp*0.001*Constants.Uf1*Constants.RhoG1)/Constants.muG1;
    return Re1;
}
public static double equation3_mG1(double Ab){
    double mG1 = 0.0;
    mG1 = Constants.RhoG1*Constants.Uf1*Ab; // kg/s
    return mG1;
}
public static double equation3_h1(double dp, double Re1){
    double h1 = 0.0;

```

```

    h1 =
    Constants.kG1/(dp*0.001)*(2+1.1*(Math.pow(Constants.Pr1,0.33333))*(Math.pow(Re1,0.6))); //
    kW/m2K
    return h1;
}

public static double equation3_Qg1(double mG1, int Tge1){
    double Qg1 = 0.0;
    Qg1 = mG1*Constants.Cp1*(Constants.Tgc1-Tge1)*0.001; //kW
    return Qg1;
}

public static double equation3_H1 (double Qg1, double dp, double Tps1, double Tge1, double h1,
double At){
    double H1 = 0.0;
    H1 = Qg1*(dp*0.001)*(Math.log((Fase3.Constants.Tpc1-Fase3.Constants.Tgc1)/(Tps1-
Tge1)))*(1/(h1*At*6*(1-Fase1.Constants.E)))*(1/((Fase3.Constants.Tpc1-Fase3.Constants.Tgc1)-
(Tps1-Tge1)));
    return H1;
}
}

package Fase4;

public class Constants {
    public final static double X1 = 0.837; // CH4
    public final static double X2 = 0.076; // C2H6
    public final static double X3 = 0.0192; // C3H8
    public final static double X4 = 0.007; // C4H10
    public final static double X5 = 0.0017; // C5H12
    public final static double X6 = 0.0008; // C6H14
    public final static double X7 = 0.0023; // CO2
    public final static double X8 = 0.002; // He
    public final static double X9 = 0.054; // N2
    public final static int M1 = 16;
    public final static int M2 = 30;
    public final static int M3 = 44;
    public final static int M4 = 58;
    public final static int M5 = 72;
    public final static int M6 = 86;
    public final static int M7 = 44;
    public final static int M8 = 4;

```

```
public final static double M9 = 28.15;
public final static double M10 = 44;
public final static double M11 = 18;
public final static double M12 = 28.15;
public final static double M13 = 4;
public final static double M14 = 32;
public final static double b4 = 126.15;
public final static double a1 = 13.39;
public final static double a2 = 2.28;
public final static double a3 = 0.84;
public final static double a4 = 0.41;
public final static double a5 = 0.12;
public final static double a6 = 0.07;
public final static double a7 = 0.10;
public final static double a8 = 0.01;
public final static double a9 = 1.52;
public final static double b1 = 18.74;
public final static double Cp10 = 1.3197; // CO2
public final static double Cp11 = 2.6042; //H2O
public final static double Cp12 = 1.241; // N2
public final static double Cp13 = 5.234; // He
public final static double Cp14 = 1.2083; // O2
public final static double mu10 = 0.00004786;
public final static double mu11 = 0.00002677;
public final static double mu12 = 0.0000413;
public final static double mu13 = 0.0000465;
public final static double mu14 = 0.0000493;
public final static double mu = 0.0000405199873422109; // 50% excesso
public final static double mu = 0.0000399376361836751; // 30% excesso
public final static double mu = 0.000040351991373418; // 40% excesso
public final static double k10 = 0.101309; //CO2
public final static double k11 = 0.180265; //H2O
public final static double k12 = 0.090912; //N2
public final static double k13 = 0.00017038; //He
public final static double k14 = 0.09839; //O2
public final static double k = 0.108690291223773; // 50% excesso
public final static double k = 0.11214316086005; // 20% excesso
public final static double k = 0.10969818513613; // 40% excesso
```



```
public final static double rhoGN = 0.8404;
public final static double PCIm = 37.91;
public final static int CalorReacao = 4484;
public final static int C = 12;
public final static int O = 16;
public final static double N = 28.15;
public final static int He = 4;
public final static int H = 1;
}
```

```
package Fase4;
```

```
public class Variables {
    public static double e;
    public static double n0Esteq;
    public static double n1Esteq;
    public static double n2Esteq;
    public static double n3Esteq;
    public static double n4Esteq;
    public static double n1;
    public static double n2;
    public static double n3;
    public static double n4;
    public static double n5;
    public static double X10;
    public static double a10;
    public static double y10;
    public static double c10;
    public static double e10;
    public static double G10;
    public static double X11;
    public static double a11;
    public static double y11;
    public static double c11;
    public static double e11;
    public static double G11;
    public static double X12;
    public static double a12;
    public static double y12;
    public static double c12;
```

```
public static double e12;
public static double G12;
public static double X13;
public static double a13;
public static double y13;
public static double c13;
public static double e13;
public static double G13;
public static double X14;
public static double a14;
public static double y14;
public static double c14;
public static double e14;
public static double G14;
public static double b2; //Somatório ni
public static double b3; // Xi*Mi
public static double Cpp;
public static double b4; // Somatório Mi
public static double b5; // Somatório MUi
public static double CalorNec;
public static double PCI;
public static double mCombDia;
public static double mComb; // kg/s (caudal de combustível)
public static double mp; // kg (Massa dos produtos)
public static double RazaoProdComb;
public static double dotMp; // kg/s (caudal de produtos)

public void definirVariaveisUtilizador(double e_user){
    e = e_user;
}

public void definirVariaveis( double n0Esteq_user, double n1Esteq_user, double n2Esteq_user,
double n3Esteq_user, double n4Esteq_user, double n1_user, double n2_user, double n3_user, double
n4_user, double n5_user, double X10_user, double X11_user, double X12_user, double X13_user,
double X14_user, double a10_user, double a11_user, double a12_user, double a13_user, double
a14_user, double y10_user, double y11_user, double y12_user, double y13_user, double y14_user,
double c10_user, double c11_user, double c12_user, double c13_user, double c14_user, double
e10_user, double e11_user, double e12_user, double e13_user, double e14_user, double G10_user,
double G11_user, double G12_user, double G13_user, double G14_user, double b2_user, double
```

```
b3_user, double Cpp_user, double b4_user, double b5_user, double CalorNec_user, double PCI_user,
double mCombDia_user, double mComb_user, double mp_user, double RazaoProdComb_user,
double dotMp_user){
    n0Esteq = n0Esteq_user;
    n1Esteq = n1Esteq_user;
    n2Esteq = n2Esteq_user;
    n3Esteq = n3Esteq_user;
    n4Esteq = n4Esteq_user;
    n1 = n1_user;
    n2 = n2_user;
    n3 = n3_user;
    n4 = n4_user;
    n5 = n5_user;
    X10 = X10_user;
    X11 = X11_user;
    X12 = X12_user;
    X13 = X13_user;
    X14 = X14_user;
    a10 = a10_user;
    a11 = a11_user;
    a12 = a12_user;
    a13 = a13_user;
    a14 = a14_user;
    y10 = y10_user;
    y11 = y11_user;
    y12 = y12_user;
    y13 = y13_user;
    y14 = y14_user;
    c10 = c10_user;
    c11 = c11_user;
    c12 = c12_user;
    c13 = c13_user;
    c14 = c14_user;
    b2 = b2_user;
    b3 = b3_user;
    Cpp = Cpp_user;
    CalorNec = CalorNec_user;
    PCI = PCI_user;
```

```
mCombDia = mCombDia_user;
mComb = mComb_user;
mp = mp_user;
RazaoProdComb = RazaoProdComb_user;
dotMp = dotMp_user;
}
public double gete() {
    return e;
}
public double getn0Esteq() {
    return n0Esteq;
}
public double getn1Esteq() {
    return n1Esteq;
}
public double getn2Esteq() {
    return n2Esteq;
}
public double getn3Esteq() {
    return n3Esteq;
}
public double getn4Esteq() {
    return n4Esteq;
}
public double getn1() {
    return n1;
}
public double getn2() {
    return n2;
}
public double getn3() {
    return n3;
}
public double getn4() {
    return n4;
}
public double getn5() {
    return n5;
}
```

```
}  
public double getX10() {  
    return X10;  
}  
public double getX11() {  
    return X11;  
}  
public double getX12() {  
    return X12;  
}  
public double getX13() {  
    return X13;  
}  
public double getX14() {  
    return X14;  
}  
public double geta10() {  
    return a10;  
}  
public double geta11() {  
    return a11;  
}  
public double geta12() {  
    return a12;  
}  
public double geta13() {  
    return a13;  
}  
public double geta14() {  
    return a14;  
}  
public double gety10() {  
    return y10;  
}  
public double gety11() {  
    return y11;  
}  
public double gety12() {
```

```
    return y12;
}
public double gety13() {
    return y13;
}
public double gety14() {
    return y14;
}
public double getc10() {
    return c10;
}
public double getc11() {
    return c11;
}
public double getc12() {
    return c12;
}
public double getc13() {
    return c13;
}
public double getc14() {
    return c14;
}
public double getb2() {
    return b2;
}
public double getb3() {
    return b3;
}
public double getb4() {
    return b4;
}
public double getb5() {
    return b5;
}
public double getCpp() {
    return Cpp;
}
```

```
public double getCalorNec() {
    return CalorNec;
}
public double getPCI() {
    return PCI;
}
public double getmCombDia() {
    return mCombDia;
}
public double getmComb() {
    return mComb;
}
public double getmp() {
    return mp;
}
public double getRazaoProdComb() {
    return RazaoProdComb;
}
public double getdotMp() {
    return dotMp;
}
}

package Fase4;
import Viewer.Resultados;
public class Equations {
    public static void equation4(){
        Resultados.resultados.append("----- Fase 4: Combustão -----
").append(System.getProperty("line.separator"));
        Variables.n1Esteq = equation4_n1Esteq(Constants.X1);
        System.out.println("A quantidade de n1Esteq é: " + Variables.n1Esteq);
        Variables.n2Esteq = equation4_n2Esteq(Constants.X1);
        System.out.println("A quantidade de n2Esteq é: " + Variables.n2Esteq);
        Variables.n0Esteq = equation4_n0Esteq(Variables.n1Esteq, Variables.n2Esteq);
        System.out.println("A Quantidade de n0Esteq é: " + Variables.n0Esteq);
        Variables.n3Esteq = equation4_n3Esteq(Variables.n0Esteq);
        System.out.println("A quantidade de n3Esteq é: " + Variables.n3Esteq);
        Variables.n4Esteq = equation4_n4Esteq(Constants.X8);
        System.out.println("A quantidade de n4Esteq é: " + Variables.n4Esteq);
```

```
Variables.n1 = equation4_n1(Variables.n1Esteq);
System.out.println("A quantidade de n1 é: " + Variables.n1);
Variables.n2 = equation4_n2(Variables.n2Esteq);
System.out.println("A quantidade de n2 é: " + Variables.n2);
Variables.n3 = equation4_n3(Variables.n0Esteq, Variables.e);
System.out.println("A quantidade de n3 é: " + Variables.n3);
Variables.n4 = equation4_n4(Constants.X8);
System.out.println("A quantidade de n4 é: " + Variables.n4);
Variables.n5 = equation4_n5(Variables.n1, Variables.n2, Variables.n0Esteq, Variables.e);
System.out.println("A quantidade de n5 é: " + Variables.n5);
Variables.b2 = equation4_b2(Variables.n1, Variables.n2, Variables.n3, Variables.n4,
Variables.n5);
System.out.println("A quantidade de b2 é: " + Variables.b2);
Variables.X10 = equation4_X10(Variables.n1, Variables.b2);
System.out.println("A quantidade de X10 é: " + Variables.X10);
Resultados.resultados.append("Fração molar do CO2 é:
").append(Variables.X10).append(System.getProperty("line.separator"));
Variables.X11 = equation4_X11(Variables.n2, Variables.b2);
System.out.println("A quantidade de X11 é: " + Variables.X11);
Resultados.resultados.append("Fração molar do H2O é:
").append(Variables.X11).append(System.getProperty("line.separator"));
Variables.X12 = equation4_X12(Variables.n3, Variables.b2);
System.out.println("A quantidade de X12 é: " + Variables.X12);
Resultados.resultados.append("Fração molar do N2 é:
").append(Variables.X12).append(System.getProperty("line.separator"));
Variables.X13 = equation4_X13(Variables.n4, Variables.b2);
System.out.println("A quantidade de X13 é: " + Variables.X13);
Resultados.resultados.append("Fração molar do He é:
").append(Variables.X13).append(System.getProperty("line.separator"));
Variables.X14 = equation4_X14(Variables.n5, Variables.b2);
System.out.println("A quantidade de X14 é: " + Variables.X14);
Resultados.resultados.append("Fração molar do O2 é:
").append(Variables.X14).append(System.getProperty("line.separator"));
Variables.a10 = equation4_a10(Variables.X10);
System.out.println("A quantidade de a10 é: " + Variables.a10);
Variables.a11 = equation4_a11(Variables.X11);
System.out.println("A quantidade de a11 é: " + Variables.a11);
Variables.a12 = equation4_a12(Variables.X12);
```



```
System.out.println("A quantidade de a12 é: " + Variables.a12);
Variables.a13 = equation4_a13(Variables.X13);
System.out.println("A quantidade de a13 é: " + Variables.a13);
Variables.a14 = equation4_a14(Variables.X14);
System.out.println("A quantidade de a14 é: " + Variables.a14);
Variables.b3 = equation4_b3(Variables.a10, Variables.a11, Variables.a12, Variables.a13,
Variables.a14);
    System.out.println("A quantidade de b3 é: " + Variables.b3);
    Variables.y10 = equation4_y10(Variables.a10, Variables.b3);
    System.out.println("A quantidade de y10 é: " + Variables.y10);
    Resultados.resultados.append("Fração mássica do CO2 é:
").append(Variables.y10).append(System.getProperty("line.separator"));
    Variables.y11 = equation4_y11(Variables.a11, Variables.b3);
    System.out.println("A quantidade de y11 é: " + Variables.y11);
    Resultados.resultados.append("Fração mássica do H2O é:
").append(Variables.y11).append(System.getProperty("line.separator"));
    Variables.y12 = equation4_y12(Variables.a12, Variables.b3);
    System.out.println("A quantidade de y12 é: " + Variables.y12);
    Resultados.resultados.append("Fração mássica do N2 é:
").append(Variables.y12).append(System.getProperty("line.separator"));
    Variables.y13 = equation4_y13(Variables.a13, Variables.b3);
    System.out.println("A quantidade de y13 é: " + Variables.y13);
    Resultados.resultados.append("Fração mássica do He é:
").append(Variables.y13).append(System.getProperty("line.separator"));
    Variables.y14 = equation4_y14(Variables.a14, Variables.b3);
    System.out.println("A quantidade de y14 é: " + Variables.y14);
    Resultados.resultados.append("Fração mássica do O2 é:
").append(Variables.y14).append(System.getProperty("line.separator"));
    Variables.c10 = equation4_c10(Variables.y10);
    System.out.println("A quantidade de c10 é: " + Variables.c10);
    Variables.c11 = equation4_c11(Variables.y11);
    System.out.println("A quantidade de c11 é: " + Variables.c11);
    Variables.c12 = equation4_c12(Variables.y12);
    System.out.println("A quantidade de c12 é: " + Variables.c12);
    Variables.c13 = equation4_c13(Variables.y13);
    System.out.println("A quantidade de c13 é: " + Variables.c13);
    Variables.c14 = equation4_c14(Variables.y14);
    System.out.println("A quantidade de c14 é: " + Variables.c14);
```

```

Variables.Cpp = equation4_Cpp(Variables.c10, Variables.c11, Variables.c12, Variables.c13,
Variables.c14);
System.out.println("A quantidade de Cpp é: " + Variables.Cpp);
Resultados.resultados.append("O calor especifico dos produtos (Cpp) é:
").append(Variables.Cpp).append(System.getProperty("line.separator"));
Variables.CalorNec = equation4_CalorNec(Fase1.Variables.quantidadeDeCaO);
System.out.println("A quantidade de CAlorNec é: " + Variables.CalorNec);
Resultados.resultados.append("O calor necessário para a reação é:
").append(Variables.CalorNec).append(System.getProperty("line.separator"));
Variables.PCI = equation4_PCI(Constants.PCIIm);
System.out.println("A quantidade de PCI é: " + Variables.PCI);
Resultados.resultados.append("O PCI é:
").append(Variables.PCI).append(System.getProperty("line.separator"));
Variables.mCombDia = equation4_mCombDia(Variables.CalorNec, Variables.PCI);
System.out.println("O caudal de Combustível por dia é (mCombDia) [kg/dia]: " +
Variables.mCombDia);
Resultados.resultados.append("O caudal de Combustível por dia (mCombDia) é [kg/dia]:
").append(Variables.mCombDia).append(System.getProperty("line.separator"));
Variables.mComb = equation4_mComb(Variables.CalorNec, Variables.PCI);
System.out.println("O caudal de Combustível necessário é (mComb) : " + Variables.mComb);
Resultados.resultados.append("O caudal de Combustível necessário (mComb) é [kg/s]:
").append(Variables.mComb).append(System.getProperty("line.separator"));
Variables.mp = equation4_mp(Variables.n1, Variables.n2, Variables.n3, Variables.n4,
Variables.n5);
System.out.println("A quantidade de mp é: " + Variables.mp);
Resultados.resultados.append("A massa dos produtos (mp) é [kg]:
").append(Variables.mp).append(System.getProperty("line.separator"));
Variables.RazaoProdComb = equation4_RazaoProdComb(Variables.mp);
System.out.println("A quantidade da Razão Produto/Combustível é: " +
Variables.RazaoProdComb);
Resultados.resultados.append("Razão Prdutos/Combustível é:
").append(Variables.RazaoProdComb).append(System.getProperty("line.separator"));
Variables.dotMp = equation4_dotMp(Variables.RazaoProdComb, Variables.mComb);
System.out.println("O caudal dos produtos é (dotMp): " + Variables.dotMp);
Resultados.resultados.append("O caudal dos produtos (dotMp) é [kg/s]:
").append(Variables.dotMp).append(System.getProperty("line.separator"));
Resultados.resultados.append("----- Fim da Fase 4 -----
").append(System.getProperty("line.separator")).append(System.getProperty("line.separator"));

```

```

}
public static double equation4_n1Esteq(double X1){
    double n1Esteq = 0.0;
    n1Esteq =
X1+2*Constants.X2+3*Constants.X3+4*Constants.X4+5*Constants.X5+6*Constants.X6+Constants.X7
;
    return n1Esteq;
}
public static double equation4_n2Esteq(double X1){
    double n2Esteq = 0.0;
    n2Esteq =
(4*X1+6*Constants.X2+8*Constants.X3+10*Constants.X4+12*Constants.X5+14*Constants.X6)/2;
    return n2Esteq;
}
public static double equation4_n0Esteq(double n1Esteq, double n2Esteq){
    double n0Esteq = 0.0;
    n0Esteq = (2*n1Esteq+n2Esteq-2*Constants.X7)/2;
    return n0Esteq;
}
public static double equation4_n3Esteq(double n0Esteq){
    double n3Esteq = 0.0;
    n3Esteq = Constants.X9+n0Esteq*3.76;
    return n3Esteq;
}
public static double equation4_n4Esteq(double X8){
    double n4Esteq = 0.0;
    n4Esteq = X8;
    return n4Esteq;
}
public static double equation4_n1(double n1Esteq){
    double n1 = 0.0;
    n1 = n1Esteq;
    return n1;
}
public static double equation4_n2(double n2Esteq){
    double n2 = 0.0;
    n2 = n2Esteq;
    return n2;
}

```

```
}  
public static double equation4_n3(double n0Esteq, double e){  
    double n3 = 0.0;  
    n3 = (n0Esteq+(1+e)*2*3.76+Constants.X9)/2;  
    return n3;  
}  
public static double equation4_n4(double X8){  
    double n4 = 0.0;  
    n4 = X8;  
    return n4;  
}  
public static double equation4_n5(double n1, double n2, double n0Esteq, double e){  
    double n5 = 0.0;  
    n5 = (n0Esteq *(1+e)*2-2*n1-n2+2*Constants.X7)/2;  
    return n5;  
}  
public static double equation4_b2(double n1, double n2, double n3, double n4, double n5){  
    double b2 = 0.0;  
    b2 = n1+n2+n3+n4+n5;  
    return b2;  
}  
public static double equation4_X10(double n1, double b2){  
    double X10 = 0.0;  
    X10 = n1/b2;  
    return X10;  
}  
public static double equation4_X11(double n2, double b2){  
    double X11 = 0.0;  
    X11 = n2/b2;  
    return X11;  
}  
public static double equation4_X12(double n3, double b2){  
    double X12 = 0.0;  
    X12 = n3/b2;  
    return X12;  
}  
public static double equation4_X13(double n4, double b2){  
    double X13 = 0.0;
```

```
X13 = n4/b2;
return X13;
}
public static double equation4_X14(double n5, double b2){
    double X14 = 0.0;
    X14 = n5/b2;
    return X14;
}
public static double equation4_a10(double X10){
    double a10 = 0.0;
    a10 = X10*Constants.M10;
    return a10;
}
public static double equation4_a11(double X11){
    double a11 = 0.0;
    a11 = X11*Constants.M11;
    return a11;
}
public static double equation4_a12(double X12){
    double a12 = 0.0;
    a12 = X12*Constants.M12;
    return a12;
}
public static double equation4_a13(double X13){
    double a13 = 0.0;
    a13 = X13*Constants.M13;
    return a13;
}
public static double equation4_a14(double X14){
    double a14 = 0.0;
    a14 = X14*Constants.M14;
    return a14;
}
public static double equation4_b3(double a10, double a11, double a12, double a13, double
a14){
    double b3 = 0.0;
    b3 = a10+a11+a12+a13+a14;
    return b3;
```

```
}  
public static double equation4_y10(double a10, double b3){  
    double y10 = 0.0;  
    y10 = a10/b3;  
    return y10;  
}  
public static double equation4_y11(double a11, double b3){  
    double y11 = 0.0;  
    y11 = a11/b3;  
    return y11;  
}  
public static double equation4_y12(double a12, double b3){  
    double y12 = 0.0;  
    y12 = a12/b3;  
    return y12;  
}  
public static double equation4_y13(double a13, double b3){  
    double y13 = 0.0;  
    y13 = a13/b3;  
    return y13;  
}  
public static double equation4_y14(double a14, double b3){  
    double y14 = 0.0;  
    y14 = a14/b3;  
    return y14;  
}  
public static double equation4_c10(double y10){  
    double c10 = 0.0;  
    c10 = y10*Constants.Cp10;  
    return c10;  
}  
public static double equation4_c11(double y11){  
    double c11 = 0.0;  
    c11 = y11*Constants.Cp11;  
    return c11;  
}  
public static double equation4_c12(double y12){  
    double c12 = 0.0;
```

```
        c12 = y12*Constants.Cp12;
        return c12;
    }
    public static double equation4_c13(double y13){
        double c13 = 0.0;
        c13 = y13*Constants.Cp13;
        return c13;
    }
    public static double equation4_c14(double y14){
        double c14 = 0.0;
        c14 = y14*Constants.Cp14;
        return c14;
    }
    public static double equation4_Cpp(double c10, double c11, double c12, double c13, double
c14){
        double Cpp = 0.0;
        Cpp = c10+c11+c12+c13+c14;
        return Cpp;
    }
    public static double equation4_CalorNec(double quantidadeDeCaO){
        double CalorNec = 0.0;
        CalorNec = quantidadeDeCaO*Constants.CalorReacao;
        return CalorNec;
    }
    public static double equation4_PCI(double PCIIm){
        double PCI = 0.0;
        PCI = PCIIm/Constants.rhoGN*1000;
        return PCI;
    }
    public static double equation4_mCombDia(double CalorNec, double PCI){
        double mCombDia = 0.0;
        mCombDia = CalorNec/PCI;
        return mCombDia;
    }
    public static double equation4_mComb(double CalorNec, double PCI){
        double mComb = 0.0;
        mComb = CalorNec/PCI/86400;
        return mComb;
    }
```

```

    }
    public static double equation4_mp(double n1, double n2, double n3, double n4, double n5){
        double mp = 0.0;
        mp =
n1*(Constants.C+2*Constants.O)+n2*(2+Constants.O)+n3*Constants.N+4*n4+n5*2*Constants.O;
        return mp;
    }
    public static double equation4_RazaoProdComb(double mp){
        double RazaoProdComb = 0.0;
        RazaoProdComb = mp/Constants.b1;
        return RazaoProdComb;
    }
    public static double equation4_dotMp(double RazaoProdComb, double mComb){
        double dotMp = 0.0;
        dotMp = RazaoProdComb*mComb;
        return dotMp;
    }
}

package Fase5;
public class Constants {
    public final static double Uf3 = 1;
    public final static double rhoG3 = 0.4571;
    public final static double Pr3 = 0.6884;
    public final static double Tge3 = 1000.0;
    public final static double Tp3 = 850.0;
    public final static double Tgs3 = 200.0;
    public final static double Tip = 800.0;
}

package Fase5;
public class Variables {
    public static int Tpe; // Temperatura da partícula à entrada do forno.
    public static double Re3;
    public static double mG3;
    public static double mCO2;
    public static double m3;
    public static double h3;
    public static double Qg3;
    public static double H3;
}

```



```
public void definirVariaveisUtilizador(int Tpe_user){
    Tpe = Tpe_user;
}

public void definirVariaveis(double Re3_user, double mG3_user, double mCO2_user, double
m3_user, double h3_user, double Qg3_user, double H3_user){
    Re3 = Re3_user;
    mG3 = mG3_user;
    mCO2 = mCO2_user;
    m3 = m3_user;
    h3 = h3_user;
    Qg3 = Qg3_user;
    H3 = H3_user;
}

public int getTpe() {
    return Tpe;
}

public double getRe3() {
    return Re3;
}

public double getmG3() {
    return mG3;
}

public double getmCO2() {
    return mCO2;
}

public double getm3() {
    return m3;
}

public double geth3() {
    return h3;
}

public double getQg3() {
    return Qg3;
}

public double getH3() {
    return H3;
}
}
```

```

package Fase5;
import Viewer.Resultados;
public class Equations {
    public static void equation5(){
        Resultados.resultados.append("----- Fase 5: Altura da zona de pré-aquecimento -----");
        ").append(System.getProperty("line.separator"));
        Variables.Re3 = equation5_Re3(Fase1.Variables.dp);
        System.out.println("A quantidade de Re3 é: " + Variables.Re3);
        Resultados.resultados.append("O número de Reynolds (Re3) é:");
        ").append(Variables.Re3).append(System.getProperty("line.separator"));
        Variables.mG3 = equation5_mG3(Fase3.Variables.Ab);
        System.out.println("A quantidade de mG3 é: " + Variables.mG3);
        Resultados.resultados.append("O caudal de gases (mG3) é [kg/s]:");
        ").append(Variables.mG3).append(System.getProperty("line.separator"));
        Variables.mCO2 = equation5_mCO2(Fase1.Variables.quantidadeDeCaO);
        System.out.println("A quantidade de mCO2 é: " + Variables.mCO2);
        Resultados.resultados.append("O caudal de dióxido de carbono (mCO2) é [kg/s]:");
        ").append(Variables.mCO2).append(System.getProperty("line.separator"));
        Variables.m3 = equation5_m3(Variables.mG3, Variables.mCO2, Fase4.Variables.dotMp);
        System.out.println("A quantidade de m3 é: " + Variables.m3);
        Resultados.resultados.append("O caudal total de gases (m3) é [kg/s]:");
        ").append(Variables.m3).append(System.getProperty("line.separator"));
        Variables.h3 = equation5_h3(Fase1.Variables.dp, Variables.Re3);
        System.out.println("A quantidade de h3 é: " + Variables.h3);
        Resultados.resultados.append("O coeficiente de transferência de calor por convecção (h3) é [W/m2K]: ");
        ").append(Variables.h3).append(System.getProperty("line.separator"));
        Variables.Qg3 = equation5_Qg3(Variables.m3, Fase4.Variables.Cpp);
        System.out.println("A quantidade de thetaG3 é: " + Variables.Qg3);
        Resultados.resultados.append("A quantidade de energia (Qg3) é [kW]:");
        ").append(Variables.Qg3).append(System.getProperty("line.separator"));
        Variables.H3 = equation5_H3(Variables.Qg3, Fase1.Variables.dp, Fase5.Variables.Tpe,
        Variables.h3, Fase3.Variables.At);
        System.out.println("A altura da zona de pré-aquecimento é (H3): " + Variables.H3);
        Resultados.resultados.append("A altura da zona de pré-aquecimento (H3) é [m]:");
        ").append(Variables.H3).append(System.getProperty("line.separator"));
        Resultados.resultados.append("----- Fim da Fase 5 -----");
        ").append(System.getProperty("line.separator")).append(System.getProperty("line.separator"));
    }
}

```

```
public static double equation5_Re3(double dp){
    double Re3 = 0.0;
    Re3 = (dp*0.001*Constants.Uf3*Constants.rhoG3)/Fase4.Constants.mu;
    return Re3;
}
public static double equation5_mG3(double Ab){
    double mG3 = 0.0;
    mG3 = Constants.rhoG3*Constants.Uf3*Ab; // kg/s
    return mG3;
}
public static double equation5_mCO2(double quantidadeDeCaO){
    double mCO2 = 0.0;
    mCO2 = Fase1.Variables.quantidadeDeCaCO3*0.44/86400; // kg/s
    return mCO2;
}
public static double equation5_m3(double mG3, double mCO2, double dotMp){
    double m3 = 0.0;
    m3 = mG3+mCO2+Fase4.Variables.dotMp;
    return m3;
}
public static double equation5_h3(double dp, double Re3){
    double h3 = 0.0;
    h3 =
(Fase4.Constants.k*0.001)/(dp*0.001)*(2+1.1*(Math.pow(Constants.Pr3,0.3333))*(Math.pow(Re3,0.6))
); // kW/m2K
    return h3;
}
public static double equation5_Qg3(double m3, double Cpp){
    double Qg3 = 0.0;
    Qg3 = m3*Fase4.Variables.Cpp*(Constants.Tge3-Constants.Tgs3)*0.001; //kW
    return Qg3;
}
public static double equation5_H3 (double Qg3, double dp, double Tpe, double h3, double At){
    double H3 = 0.0;
    H3 = Qg3*(dp*0.001)*(Math.log((Fase5.Constants.Tgs3-Tpe)/(Fase5.Constants.Tge3-
Fase5.Constants.Tp3)))*(1/(h3*At*6*(1-Fase1.Constants.E)))*(1/((Fase5.Constants.Tgs3-Tpe)-
(Fase5.Constants.Tge3-Fase5.Constants.Tp3)))); //m
    return H3;
}
```

```
}  
}  
package Fase6;  
public class Variables {  
    public static double Hzona;  
    public void definirVariaveis(double Hzona_user){  
        Hzona = Hzona_user;  
    }  
    public double getHzona() {  
        return Hzona;  
    }  
}  
package Fase6;  
import Viewer.Resultados;  
public class Equations {  
    public static boolean equation6() {  
        Resultados.resultados.append("----- Fase 6: Verificação da altura do forno -----  
").append(System.getProperty("line.separator"));  
        Variables.Hzona = equation6_Hzona(Fase2.Variables.H2, Fase3.Variables.H1,  
Fase5.Variables.H3);  
        System.out.println("A quantidade de Hzona é: " + Variables.Hzona);  
        Resultados.resultados.append("O somatório da altura das zonas (Hzona) é [m]:  
").append(Variables.Hzona).append(System.getProperty("line.separator"));  
        if (Variables.Hzona <= Fase1.Variables.H){  
            Variables.Hzona = equation6_Hzona(Fase2.Variables.H2, Fase3.Variables.H1,  
Fase5.Variables.H3);  
            System.out.println("A quantidade de Hzona é: " + Variables.Hzona);  
            Resultados.resultados.append("----- Fim da Fase 6: COM SUCESSO -----  
").append(System.getProperty("line.separator")).append(System.getProperty("line.separator"));  
            return true;  
        }  
        else{  
            System.out.println("ERROR");  
            Resultados.resultados.append("----- Fim da Fase 6 SEM SUCESSO: O somatório da altura  
das zonas é maior do que a altura do forno (H) -----  
").append(System.getProperty("line.separator")).append(System.getProperty("line.separator"));  
            return false;  
        }  
    }  
}
```

```

}
public static double equation6_Hzona(double H2, double H1, double H3){
    double Hzona = 0.0;
    Hzona = H2+H1+H3;
    return Hzona;
}
}
package Fase7;
public class Constants {
    public final static double Kr1 = 1.4; // Tijolo refratário W/mK
    public final static double Kr2 = 0.57; // betão isolante W/mK
    public final static double Lr1 = 0.2;
    public final static double Lr2 = 0.6;
    public final static double rhoG2 = 0.30101;
    public final static double Uf2 = 1.0;
    public final static int Tpar1 = 100;
    public final static int Ti = 1200;
    public final static double Emi = 0.8; // emissividade
    public final static double sigma = 0.000000056697;
    public final static double g = 9.81; // m2/s
}
package Fase7;
public class Variables {
    public static int Tamb;
    public static double r1;
    public static double r2;
    public static double Lc;
    public static double D;
    public static double Df;
    public static double esp; // espessura das paredes
    public static double Re2;
    public static double hi;
    public static double Tpar2;
    public static double Tpar3;
    public static double Tpar4;
    public static double Tpar5;
    public static double Tpar; // Temperatura da parede 6ª iteração (última)
    public static double Tf1; // Temperatura do filme 1ª iteração

```

```
public static double Tf2;
public static double Tf3;
public static double Tf4;
public static double Tf5;
public static double Tf; // Temperatura do filme 6ª iteração (última)
public static double Kamb1; // Condutibilidade térmica 1ª iteração
public static double Kamb2;
public static double Kamb3;
public static double Kamb4;
public static double Kamb5;
public static double Kamb; // Condutibilidade térmica 6ª iteração (última)
public static double ViscoCin1; // Viscosidade Cinemática 1ª iteração
public static double ViscoCin2;
public static double ViscoCin3;
public static double ViscoCin4;
public static double ViscoCin5;
public static double ViscoCin; // Viscosidade Cinemática 6ª iteração (última)
public static double beta1; // Beta 1ª iteração
public static double beta2;
public static double beta3;
public static double beta4;
public static double beta5;
public static double beta; // Beta 6ª iteração (última)
public static double Pr1; // Prandtl 1ª iteração
public static double Pr2;
public static double Pr3;
public static double Pr4;
public static double Pr5;
public static double Pr; // Prandtl 6ª iteração (última)
public static double Gr1; // Grashof 1ª iteração
public static double Gr2;
public static double Gr3;
public static double Gr4;
public static double Gr5;
public static double Gr; // Grashof 6ª iteração (última)
public static double Ra1; // Rayleigh 1ª iteração
public static double Ra2;
public static double Ra3;
```

```
public static double Ra4;
public static double Ra5;
public static double Ra; // Rayleigh 6ª iteração (última)
public static double hamb1; // Coeficiente de convecção do ambiente 1ª iteração
public static double hamb2;
public static double hamb3;
public static double hamb4;
public static double hamb5;
public static double hamb;
public static double Nu1;
public static double Nu2;
public static double Nu3;
public static double Nu4;
public static double Nu5;
public static double Nu; //Nusselt 6ª iteração (última)
public static double hef1;
public static double hef2;
public static double hef3;
public static double hef4;
public static double hef5;
public static double hef; // Coeficiente global 6ª iteração (última)
public static double Ri; // Resistência térmica (interior) 6ª iteração (última)
public static double Rr1; // Resistência térmica (Refratário 1) 6ª iteração (última)
public static double Rr2; // Resistência térmica (Refratário 2) 6ª iteração (última)
public static double Rex1; // Resistência térmica (Exterior) 1ª iteração
public static double Rex2;
public static double Rex3;
public static double Rex4;
public static double Rex5;
public static double Rex;
public static double Rt1; // Resistência térmica total 1ª iteração
public static double Rt2;
public static double Rt3;
public static double Rt4;
public static double Rt5;
public static double Rt;
public static double q1; // Fluxo de calor 1ª iteração
public static double q2; // Fluxo de calor 2ª iteração
```

```
public static double q3; // Fluxo de calor 3ª iteração
public static double q4; // Fluxo de calor 4ª iteração
public static double q5; // Fluxo de calor 5ª iteração
public static double q; // Fluxo de calor 6ª iteração (última)
public static double Tpi1; // Temperatura da parede interior 1ª iteração
public static double Tpi2; // Temperatura da parede interior 2ª iteração
public static double Tpi3; // Temperatura da parede interior 3ª iteração
public static double Tpi4; // Temperatura da parede interior 4ª iteração
public static double Tpi5; // Temperatura da parede interior 5ª iteração
public static double Tpi; // Temperatura da parede interior 6ª iteração (última)
public static double Tr11; // Temperatura no Refratário 1ª iteração
public static double Tr12; // Temperatura no Refratário 2ª iteração
public static double Tr13; // Temperatura no Refratário 3ª iteração
public static double Tr14; // Temperatura no Refratário 4ª iteração
public static double Tr15; // Temperatura no Refratário 5ª iteração
public static double Tr1; // Temperatura no Refratário 6ª iteração (última)
public static double Tpex1; // Temperatura da parede exterior 1ª iteração
public static double Tpex2; // Temperatura da parede exterior 2ª iteração
public static double Tpex3; // Temperatura da parede exterior 3ª iteração
public static double Tpex4; // Temperatura da parede exterior 4ª iteração
public static double Tpex5; // Temperatura da parede exterior 5ª iteração
public static double Tpex; // Temperatura da parede exterior 6ª iteração (última)
public void definirVariaveisUtilizador(int Tamb_user){
    Tamb = Tamb_user;
}

public void definirVariaveis(double r1_user, double r2_user, double Lc_user, double D_user,
double Df_user, double esp_user, double Re2_user, double hi_user, double Tpar2_user, double
Tpar3_user, double Tpar4_user, double Tpar5_user, double Tpar_user, double Tf1_user, double
Tf2_user, double Tf3_user, double Tf4_user, double Tf5_user, double Tf_user, double Kamb1_user,
double Kamb2_user, double Kamb3_user, double Kamb4_user, double Kamb5_user, double
Kamb_user, double ViscoCin1_user, double ViscoCin2_user, double ViscoCin3_user, double
ViscoCin4_user, double ViscoCin5_user, double ViscoCin_user, double beta1_user, double
beta2_user, double beta3_user, double beta4_user, double beta5_user, double beta_user, double
Pr1_user, double Pr2_user, double Pr3_user, double Pr4_user, double Pr5_user, double Pr_user,
double Gr1_user, double Gr2_user, double Gr3_user, double Gr4_user, double Gr5_user, double
Gr_user, double Ra1_user, double Ra2_user, double Ra3_user, double Ra4_user, double Ra5_user,
double Ra_user, double hamb_user, double Nu1_user, double Nu2_user, double Nu3_user, double
Nu4_user, double Nu5_user, double Nu_user, double hamb1_user, double hamb2_user, double
```



```
hamb3_user, double hamb4_user, double hamb5_user, double hef1_user, double hef2_user, double
hef3_user, double hef4_user, double hef5_user, double hef_user, double Ri_user, double Rr1_user,
double Rr2_user, double Rex1_user, double Rex2_user, double Rex3_user, double Rex4_user,
double Rex5_user, double Rex_user, double Rt1_user, double Rt2_user, double Rt3_user, double
Rt4_user, double Rt5_user, double Rt_user, double q1_user, double q2_user, double q3_user, double
q4_user, double q5_user, double q_user, double Tpi1_user, double Tpi2_user, double Tpi3_user,
double Tpi4_user, double Tpi5_user, double Tpi_user, double Tr11_user, double Tr12_user, double
Tr13_user, double Tr14_user, double Tr15_user, double Tr1_user, double Tpex1_user, double
Tpex2_user, double Tpex3_user, double Tpex4_user, double Tpex5_user, double Tpex_user){
    r1 = r1_user;
    r2 = r2_user;
    Lc = Lc_user;
    D = D_user;
    Df = Df_user;
    esp = esp_user;
    Re2 = Re2_user;
    hi = hi_user;
    Tpar2 = Tpar2_user;
    Tpar3 = Tpar3_user;
    Tpar4 = Tpar4_user;
    Tpar5 = Tpar5_user;
    Tpar = Tpar_user;
    Tf1 = Tf1_user;
    Tf2 = Tf2_user;
    Tf3 = Tf3_user;
    Tf4 = Tf4_user;
    Tf5 = Tf5_user;
    Tf = Tf_user;
    Kamb1 = Kamb1_user;
    Kamb2 = Kamb2_user;
    Kamb3 = Kamb3_user;
    Kamb4 = Kamb4_user;
    Kamb5 = Kamb5_user;
    Kamb = Kamb_user;
    ViscoCin1 = ViscoCin1_user;
    ViscoCin2 = ViscoCin2_user;
    ViscoCin3 = ViscoCin3_user;
    ViscoCin4 = ViscoCin4_user;
```

```
ViscoCin5 = ViscoCin5_user;  
ViscoCin = ViscoCin_user;  
beta1 = beta1_user;  
beta2 = beta2_user;  
beta3 = beta3_user;  
beta4 = beta4_user;  
beta5 = beta5_user;  
beta = beta_user;  
Pr1 = Pr1_user;  
Pr2 = Pr2_user;  
Pr3 = Pr3_user;  
Pr4 = Pr4_user;  
Pr5 = Pr5_user;  
Pr = Pr_user;  
Gr1 = Gr1_user;  
Gr2 = Gr2_user;  
Gr3 = Gr3_user;  
Gr4 = Gr4_user;  
Gr5 = Gr5_user;  
Gr = Gr_user;  
Ra1 = Ra1_user;  
Ra2 = Ra2_user;  
Ra3 = Ra3_user;  
Ra4 = Ra4_user;  
Ra5 = Ra5_user;  
Ra = Ra_user;  
Nu1 = Nu1_user;  
Nu2 = Nu2_user;  
Nu3 = Nu3_user;  
Nu4 = Nu4_user;  
Nu5 = Nu5_user;  
Nu = Nu_user;  
hamb1 = hamb1_user;  
hamb2 = hamb2_user;  
hamb3 = hamb3_user;  
hamb4 = hamb4_user;  
hamb5 = hamb5_user;  
hamb = hamb_user;
```

```
hef1 = hef1_user;
hef2 = hef2_user;
hef3 = hef3_user;
hef4 = hef4_user;
hef5 = hef5_user;
hef = hef_user;
Ri = Ri_user;
Rr1 = Rr1_user;
Rr2 = Rr2_user;
Rex1 = Rex1_user;
Rex2 = Rex2_user;
Rex3 = Rex3_user;
Rex4 = Rex4_user;
Rex5 = Rex5_user;
Rex = Rex_user;
Rt1 = Rt1_user;
Rt2 = Rt2_user;
Rt3 = Rt3_user;
Rt4 = Rt4_user;
Rt5 = Rt5_user;
Rt = Rt_user;
q1 = q1_user;
q2 = q2_user;
q3 = q3_user;
q4 = q4_user;
q5 = q5_user;
q = q_user;
Tpi1 = Tpi1_user;
Tpi2 = Tpi2_user;
Tpi3 = Tpi3_user;
Tpi4 = Tpi4_user;
Tpi5 = Tpi5_user;
Tpi = Tpi_user;
Tr11 = Tr11_user;
Tr12 = Tr12_user;
Tr13 = Tr13_user;
Tr14 = Tr14_user;
Tr15 = Tr15_user;
```

```
Tr1 = Tr1_user;
Tpex1 = Tpex1_user;
Tpex2 = Tpex2_user;
Tpex3 = Tpex3_user;
Tpex4 = Tpex4_user;
Tpex5 = Tpex5_user;
Tpex = Tpex_user;
}
public int getTamb() {
    return Tamb;
}
public double getr1() {
    return r1;
}
public double getr2() {
    return r2;
}
public double getLc() {
    return Lc;
}
public double getD() {
    return D;
}
public double getDf() {
    return Df;
}
public double getesp() {
    return esp;
}
public double getRe2() {
    return Re2;
}
public double gethi() {
    return hi;
}
public double getTpar2() {
    return Tpar2;
}
```

```
public double getTpar3() {  
    return Tpar3;  
}  
public double getTpar4() {  
    return Tpar4;  
}  
public double getTpar5() {  
    return Tpar5;  
}  
public double getTpar() {  
    return Tpar;  
}  
public double getTf1() {  
    return Tf1;  
}  
public double getTf2() {  
    return Tf2;  
}  
public double getTf3() {  
    return Tf3;  
}  
public double getTf4() {  
    return Tf4;  
}  
public double getTf5() {  
    return Tf5;  
}  
public double getTf() {  
    return Tf;  
}  
public double getKamb1() {  
    return Kamb1;  
}  
public double getKamb2() {  
    return Kamb2;  
}  
public double getKamb3() {  
    return Kamb3;  
}
```

```
}  
public double getKamb4() {  
    return Kamb4;  
}  
public double getKamb5() {  
    return Kamb5;  
}  
public double getKamb() {  
    return Kamb;  
}  
public double getViscoCin1() {  
    return ViscoCin1;  
}  
public double getViscoCin2() {  
    return ViscoCin2;  
}  
public double getViscoCin3() {  
    return ViscoCin3;  
}  
public double getViscoCin4() {  
    return ViscoCin4;  
}  
public double getViscoCin5() {  
    return ViscoCin5;  
}  
public double getViscoCin() {  
    return ViscoCin;  
}  
public double getbeta1() {  
    return beta1;  
}  
public double getbeta2() {  
    return beta2;  
}  
public double getbeta3() {  
    return beta3;  
}  
public double getbeta4() {
```

```
    return beta4;
}
public double getbeta5() {
    return beta5;
}
public double getbeta() {
    return beta;
}
public double getPr1() {
    return Pr1;
}
public double getPr2() {
    return Pr2;
}
public double getPr3() {
    return Pr3;
}
public double getPr4() {
    return Pr4;
}
public double getPr5() {
    return Pr5;
}
public double getPr() {
    return Pr;
}
public double getGr1() {
    return Gr1;
}
public double getGr2() {
    return Gr2;
}
public double getGr3() {
    return Gr3;
}
public double getGr4() {
    return Gr4;
}
```

```
public double getGr5() {  
    return Gr5;  
}  
public double getGr() {  
    return Gr;  
}  
public double getRa1() {  
    return Ra1;  
}  
public double getRa2() {  
    return Ra2;  
}  
public double getRa3() {  
    return Ra3;  
}  
public double getRa4() {  
    return Ra4;  
}  
public double getRa5() {  
    return Ra5;  
}  
public double getRa() {  
    return Ra;  
}  
public double getNu1() {  
    return Nu1;  
}  
public double getNu2() {  
    return Nu2;  
}  
public double getNu3() {  
    return Nu3;  
}  
public double getNu4() {  
    return Nu;  
}  
public double getNu5() {  
    return Nu5;  
}
```



```
}  
public double getNu() {  
    return Nu;  
}  
public double gethamb1() {  
    return hamb1;  
}  
public double gethamb2() {  
    return hamb2;  
}  
public double gethamb3() {  
    return hamb3;  
}  
public double gethamb4() {  
    return hamb4;  
}  
public double gethamb5() {  
    return hamb5;  
}  
public double gethamb() {  
    return hamb;  
}  
public double gethef1() {  
    return hef1;  
}  
public double gethef2() {  
    return hef2;  
}  
public double gethef3() {  
    return hef3;  
}  
public double gethef4() {  
    return hef4;  
}  
public double gethef5() {  
    return hef5;  
}  
public double gethef() {
```

```
    return hef;
}
public double getRi() {
    return Ri;
}
public double getRr1() {
    return Rr1;
}
public double getRr2() {
    return Rr2;
}
public double getRex1() {
    return Rex1;
}
public double getRex2() {
    return Rex2;
}
public double getRex3() {
    return Rex3;
}
public double getRex4() {
    return Rex4;
}
public double getRex5() {
    return Rex5;
}
public double getRex() {
    return Rex;
}
public double getRt1() {
    return Rt1;
}
public double getRt2() {
    return Rt2;
}
public double getRt3() {
    return Rt3;
}
```

```
public double getRt4() {  
    return Rt4;  
}  
public double getRt5() {  
    return Rt5;  
}  
public double getRt() {  
    return Rt;  
}  
public double getq1() {  
    return q1;  
}  
public double getq2() {  
    return q2;  
}  
public double getq3() {  
    return q3;  
}  
public double getq4() {  
    return q4;  
}  
public double getq5() {  
    return q5;  
}  
public double getq() {  
    return q;  
}  
public double getTpi1() {  
    return Tpi1;  
}  
public double getTpi2() {  
    return Tpi2;  
}  
public double getTpi3() {  
    return Tpi3;  
}  
public double getTpi4() {  
    return Tpi4;  
}
```

```
}  
public double getTpi5() {  
    return Tpi5;  
}  
public double getTpi() {  
    return Tpi;  
}  
public double getTr11() {  
    return Tr11;  
}  
public double getTr12() {  
    return Tr12;  
}  
public double getTr13() {  
    return Tr13;  
}  
public double getTr14() {  
    return Tr14;  
}  
public double getTr15() {  
    return Tr15;  
}  
public double getTr1() {  
    return Tr1;  
}  
public double getTpex1() {  
    return Tpex1;  
}  
public double getTpex2() {  
    return Tpex2;  
}  
public double getTpex3() {  
    return Tpex3;  
}  
public double getTpex4() {  
    return Tpex4;  
}  
public double getTpex5() {
```

```

    return Tpex5;
}
public double getTpex() {
    return Tpex;
}
}
package Fase7;
import Viewer.Resultados;
public class Equations {
    public static boolean equation7() {
        Resultados.resultados.append("----- Fase 7: Análise Transferência de calor -----");
        ").append(System.getProperty("line.separator"));
        Variables.r1 = equation7_r1(Fase1.Variables.R);
        System.out.println("A quantidade de r1 é: " + Variables.r1);
        Variables.r2 = equation7_r2(Variables.r1);
        System.out.println("A quantidade de r2 é: " + Variables.r2);
        Variables.Lc = equation7_Lc(Fase1.Variables.H);
        System.out.println("A quantidade de Lc é: " + Variables.Lc);
        Variables.D = equation7_D(Variables.r2);
        System.out.println("A quantidade de D é: " + Variables.D);
        Resultados.resultados.append("O diâmetro externo do forno (D) é [m]:");
        ").append(Variables.D).append(System.getProperty("line.separator"));
        Variables.esp = equation7_esp(Variables.D, Fase1.Variables.R);
        System.out.println("A espessura das paredes é: " + Variables.esp);
        Resultados.resultados.append("A espessura da parede (esp) é [m]:");
        ").append(Variables.esp).append(System.getProperty("line.separator"));
        Variables.Re2 = equation7_Re2(Fase1.Variables.dp);
        System.out.println("A quantidade de Re2 é: " + Variables.Re2);
        Resultados.resultados.append("O número de Reynolds (Re2) é:");
        ").append(Variables.Re2).append(System.getProperty("line.separator"));
        Variables.hi = equation7_hi(Variables.Re2);
        System.out.println("A quantidade de hi é: " + Variables.hi);
        Resultados.resultados.append("O coeficiente de transferência de calor no interior do forno (hi) é [W/m2K]: ");
        ").append(Variables.hi).append(System.getProperty("line.separator"));
        Variables.Tf1 = equation7_Tf1(Variables.Tamb);
        System.out.println("A quantidade de Tf1 é: " + Variables.Tf1);
        Resultados.resultados.append("A temperatura do filme da 1ª iteração (Tf1) é [°C]:");
        ").append(Variables.Tf1).append(System.getProperty("line.separator"));
    }
}

```

```

Variables.Kamb1 = equation7_Kamb1(Variables.Tf1);
System.out.println("A quantidade de Kamb1 é: " + Variables.Kamb1);
Resultados.resultados.append("Condutividade térmica do ar ambiente 1ª iteração (Kamb1) é
[W/mK]: ").append(Variables.Kamb1).append(System.getProperty("line.separator"));
Variables.ViscoCin1 = equation7_ViscoCin1(Variables.Tf1);
System.out.println("A quantidade de ViscoCin1 é: " + Variables.ViscoCin1);
Resultados.resultados.append("A viscosidade cinemática do ar da 1ª iteração (ViscoCin1) é
[m2/s]: ").append(Variables.ViscoCin1).append(System.getProperty("line.separator"));
Variables.beta1 = equation7_beta1(Variables.Tf1);
System.out.println("A quantidade de beta1 é: " + Variables.beta1);
Resultados.resultados.append("Coeficiente de expansão volumétrica da 1ª iteração (beta1) é
[1/K]: ").append(Variables.beta1).append(System.getProperty("line.separator"));
Variables.Pr1 = equation7_Pr1(Variables.Tf1);
System.out.println("A quantidade de Pr1 é: " + Variables.Pr1);
Resultados.resultados.append("O número de Prandtl da 1ª iteração (Pr1) é:
").append(Variables.Pr1).append(System.getProperty("line.separator"));
Variables.Gr1 = equation7_Gr1(Variables.beta1, Variables.Tamb, Variables.Lc,
Variables.ViscoCin1);
System.out.println("A quantidade de Gr1 é: " + Variables.Gr1);
Resultados.resultados.append("O número de Grashof da 1ª iteração (Gr1) é:
").append(Variables.Gr1).append(System.getProperty("line.separator"));
Variables.Df = equation7_Df(Variables.Lc, Variables.Gr1);
System.out.println("A quantidade de Df é: " + Variables.Df);
Variables.Ra1 = equation7_Ra1(Variables.Gr1, Variables.Pr1);
System.out.println("A quantidade de Ra1 é: " + Variables.Ra1);
Resultados.resultados.append("O número de Rayleigh da 1ª iteração (Ra1) é:
").append(Variables.Ra1).append(System.getProperty("line.separator"));
Variables.Nu1 = equation7_Nu1(Variables.Ra1, Variables.Pr1);
System.out.println("A quantidade de Nu1 é: " + Variables.Nu1);
Resultados.resultados.append("O número de Nusselt da 1ª iteração (Nu1) é:
").append(Variables.Nu1).append(System.getProperty("line.separator"));
Variables.hamb1 = equation7_hamb1(Variables.Nu1, Variables.Kamb1, Variables.Lc);
System.out.println("A quantidade de hamb1 é: " + Variables.hamb1);
Resultados.resultados.append("O coeficiente de convecção do ambiente da 1ª iteração (hamb1)
é [W/m2K]: ").append(Variables.hamb1).append(System.getProperty("line.separator"));
Variables.hef1 = equation7_hef1(Variables.hamb1, Variables.Tamb);
System.out.println("A quantidade de hef1 é: " + Variables.hef1);

```

```

Resultados.resultados.append("O coeficiente de global (efetivo) da 1ª iteração (hef1) é [W/m2K]:
").append(Variables.hef1).append(System.getProperty("line.separator"));
Variables.Ri = equation7_Ri(Variables.hi, Variables.Lc);
System.out.println("A quantidade de Ri é: " + Variables.Ri);
Variables.Rr1 = equation7_Rr1(Variables.r1, Fase1.Variables.R, Variables.Lc);
System.out.println("A quantidade de Rr1 é: " + Variables.Rr1);
Variables.Rr2 = equation7_Rr2(Variables.r1, Variables.r2, Variables.Lc);
System.out.println("A quantidade de Rr2 é: " + Variables.Rr2);
Variables.Rex1 = equation7_Rex1(Variables.hef1, Variables.Lc);
System.out.println("A quantidade de Rex1 é: " + Variables.Rex1);
Variables.Rt1 = equation7_Rt1(Variables.Ri, Variables.Rr1, Variables.Rr2, Variables.Rex1);
System.out.println("A quantidade de Rt1 é: " + Variables.Rt1);
Resultados.resultados.append("Resistência térmica total da 1ª iteração (Rt1) é [m2/W K]:
").append(Variables.Rt1).append(System.getProperty("line.separator"));
Variables.q1 = equation7_q1(Variables.Tamb, Variables.Rt1);
System.out.println("A quantidade de q1 é: " + Variables.q1);
Resultados.resultados.append("O fluxo de calor da 1ª iteração (q1) é [W/m2]:
").append(Variables.q1).append(System.getProperty("line.separator"));
Variables.Tpi1 = equation7_Tpi(Variables.q1, Variables.Ri);
System.out.println("A quantidade de Tpi1 é: " + Variables.Tpi1);
Variables.Tr11 = equation7_Tr11(Variables.q1, Variables.Rr1, Variables.Tpi1);
System.out.println("A quantidade de Tr11 é: " + Variables.Tr11);
Variables.Tpex1 = equation7_Tpex1(Variables.q1, Variables.Rr2, Variables.Tr11);
System.out.println("A quantidade de Tpex1 é: " + Variables.Tpex1);
Resultados.resultados.append("A temperatura da parede exterior na 1ª iteração (Tpex1) é [°C]:
").append(Variables.Tpex1).append(System.getProperty("line.separator"));
Variables.Tpar2 = equation7_Tpar2(Variables.Tpex1);
System.out.println("A quantidade de Tpar2 é: " + Variables.Tpar2);
Variables.Tf2 = equation7_Tf2(Variables.Tpar2, Variables.Tamb);
System.out.println("A quantidade de Tf2 é: " + Variables.Tf2);
Resultados.resultados.append("A temperatura do filme da 2ª iteração (Tf2) é [°C]:
").append(Variables.Tf2).append(System.getProperty("line.separator"));
Variables.Kamb2 = equation7_Kamb2(Variables.Tf2);
System.out.println("A quantidade de Kamb2 é: " + Variables.Kamb2);
Resultados.resultados.append("Condutividade térmica do ar ambiente da 2ª iteração (Kamb2) é
[W/mK]: ").append(Variables.Kamb2).append(System.getProperty("line.separator"));
Variables.ViscoCin2 = equation7_ViscoCin2(Variables.Tf2);
System.out.println("A quantidade de ViscoCin2 é: " + Variables.ViscoCin2);

```

```

Resultados.resultados.append("A viscosidade cinemática do ar da 2ª iteração (ViscoCin2) é
[m2/s]: ").append(Variables.ViscoCin2).append(System.getProperty("line.separator"));
Variables.beta2 = equation7_beta2(Variables.Tf2);
System.out.println("A quantidade de beta2 é: " + Variables.beta2);
Resultados.resultados.append("Coeficiente de expansão volumétrica da 2ª iteração (beta2) é
[1/K]: ").append(Variables.beta2).append(System.getProperty("line.separator"));
Variables.Pr2 = equation7_Pr2(Variables.Tf2);
System.out.println("A quantidade de Pr2 é: " + Variables.Pr2);
Resultados.resultados.append("O número de Prandtl da 2ª iteração (Pr2) é:
").append(Variables.Pr2).append(System.getProperty("line.separator"));
Variables.Gr2 = equation7_Gr2(Variables.beta2, Variables.Tpar2, Variables.Tamb, Variables.Lc,
Variables.ViscoCin2);
System.out.println("A quantidade de Gr2 é: " + Variables.Gr2);
Resultados.resultados.append("O número de Grashof da 2ª iteração (Gr2) é:
").append(Variables.Gr2).append(System.getProperty("line.separator"));
Variables.Ra2 = equation7_Ra2(Variables.Gr2, Variables.Pr2);
System.out.println("A quantidade de Ra2 é: " + Variables.Ra2);
Resultados.resultados.append("O número de Rayleigh da 2ª iteração (Ra2) é:
").append(Variables.Ra2).append(System.getProperty("line.separator"));
Variables.Nu2 = equation7_Nu2(Variables.Ra2, Variables.Pr2);
System.out.println("A quantidade de Nu2 é: " + Variables.Nu2);
Resultados.resultados.append("O número de Nusselt da 2ª iteração (Nu2) é:
").append(Variables.Nu2).append(System.getProperty("line.separator"));
Variables.hamb2 = equation7_hamb2(Variables.Nu2, Variables.Kamb2, Variables.Lc);
System.out.println("A quantidade de hamb2 é: " + Variables.hamb2);
Resultados.resultados.append("O coeficiente de convecção do ambiente da 2ª iteração (hamb2)
é [W/m2K]: ").append(Variables.hamb2).append(System.getProperty("line.separator"));
Variables.hef2 = equation7_hef2(Variables.hamb2, Variables.Tpar2, Variables.Tamb);
System.out.println("A quantidade de hef2 é: " + Variables.hef2);
Resultados.resultados.append("O coeficiente de global (efetivo) da 2ª iteração (hef2) é [W/m2K]:
").append(Variables.hef2).append(System.getProperty("line.separator"));
Variables.Rex2 = equation7_Rex2(Variables.hef2, Variables.Lc);
System.out.println("A quantidade de Rex2 é: " + Variables.Rex2);
Variables.Rt2 = equation7_Rt2(Variables.Ri, Variables.Rr1, Variables.Rr2, Variables.Rex2);
System.out.println("A quantidade de Rt2 é: " + Variables.Rt2);
Resultados.resultados.append("Resistência térmica total da 2ª iteração (Rt2) é [m2/W K]:
").append(Variables.Rt2).append(System.getProperty("line.separator"));
Variables.q2 = equation7_q2(Variables.Tamb, Variables.Rt2);

```



```
System.out.println("A quantidade de q2 é: " + Variables.q2);
Resultados.resultados.append("O fluxo de calor da 2ª iteração (q2) é [W/m2]:
").append(Variables.q2).append(System.getProperty("line.separator"));
Variables.Tpi2 = equation7_Tpi2(Variables.q2, Variables.Ri);
System.out.println("A quantidade de Tpi2 é: " + Variables.Tpi2);
Variables.Tr12 = equation7_Tr12(Variables.q2, Variables.Rr1, Variables.Tpi2);
System.out.println("A quantidade de Tr12 é: " + Variables.Tr12);
Variables.Tpex2 = equation7_Tpex2(Variables.q2, Variables.Rr2, Variables.Tr12);
System.out.println("A quantidade de Tpex2 é: " + Variables.Tpex2);
Resultados.resultados.append("A temperatura da parede exterior na 2ª iteração (Tpex2) é [°C]:
").append(Variables.Tpex2).append(System.getProperty("line.separator"));
Variables.Tpar3 = equation7_Tpar3(Variables.Tpex2);
System.out.println("A quantidade de Tpar3 é: " + Variables.Tpar3);
Variables.Tf3 = equation7_Tf3(Variables.Tpar3, Variables.Tamb);
System.out.println("A quantidade de Tf3 é: " + Variables.Tf3);
Resultados.resultados.append("A temperatura do filme da 3ª iteração (Tf3) é [°C]:
").append(Variables.Tf3).append(System.getProperty("line.separator"));
Variables.Kamb3 = equation7_Kamb3(Variables.Tf3);
System.out.println("A quantidade de Kamb3 é: " + Variables.Kamb3);
Resultados.resultados.append("Condutividade térmica do ar ambiente da 3ª iteração (Kamb3) é
[W/mK]: ").append(Variables.Kamb3).append(System.getProperty("line.separator"));
Variables.ViscoCin3 = equation7_ViscoCin3(Variables.Tf3);
System.out.println("A quantidade de ViscoCin3 é: " + Variables.ViscoCin3);
Resultados.resultados.append("A viscosidade cinemática do ar da 3ª iteração (ViscoCin3) é
[m2/s]: ").append(Variables.ViscoCin3).append(System.getProperty("line.separator"));
Variables.beta3 = equation7_beta3(Variables.Tf3);
System.out.println("A quantidade de beta3 é: " + Variables.beta3);
Resultados.resultados.append("Coeficiente de expansão volumétrica da 3ª iteração (beta3) é
[1/K]: ").append(Variables.beta3).append(System.getProperty("line.separator"));
Variables.Pr3 = equation7_Pr3(Variables.Tf3);
System.out.println("A quantidade de Pr3 é: " + Variables.Pr3);
Resultados.resultados.append("O número de Prandtl da 3ª iteração (Pr3) é:
").append(Variables.Pr3).append(System.getProperty("line.separator"));
Variables.Gr3 = equation7_Gr3(Variables.beta3, Variables.Tpar3, Variables.Tamb, Variables.Lc,
Variables.ViscoCin3);
System.out.println("A quantidade de Gr3 é: " + Variables.Gr3);
Resultados.resultados.append("O número de Grashof da 3ª iteração (Gr3) é:
").append(Variables.Gr3).append(System.getProperty("line.separator"));
```

```

Variables.Ra3 = equation7_Ra3(Variables.Gr3, Variables.Pr3);
System.out.println("A quantidade de Ra3 é: " + Variables.Ra3);
Resultados.resultados.append("O número de Rayleigh da 3ª iteração (Ra3) é:
").append(Variables.Ra3).append(System.getProperty("line.separator"));
Variables.Nu3 = equation7_Nu3(Variables.Ra3, Variables.Pr3);
System.out.println("A quantidade de Nu3 é: " + Variables.Nu3);
Resultados.resultados.append("O número de Nusselt da 3ª iteração (Nu3) é:
").append(Variables.Nu3).append(System.getProperty("line.separator"));
Variables.hamb3 = equation7_hamb3(Variables.Nu3, Variables.Kamb3, Variables.Lc);
System.out.println("A quantidade de hamb3 é: " + Variables.hamb3);
Resultados.resultados.append("O coeficiente de convecção do ambiente da 3ª iteração (hamb3)
é [W/m2K]: ").append(Variables.hamb3).append(System.getProperty("line.separator"));
Variables.hef3 = equation7_hef3(Variables.hamb3, Variables.Tpar3, Variables.Tamb);
System.out.println("A quantidade de hef3 é: " + Variables.hef3);
Resultados.resultados.append("O coeficiente de global (efetivo) da 3ª iteração (hef3) é [W/m2K]:
").append(Variables.hef3).append(System.getProperty("line.separator"));
Variables.Rex3 = equation7_Rex3(Variables.hef3, Variables.Lc);
System.out.println("A quantidade de Rex3 é: " + Variables.Rex3);
Variables.Rt3 = equation7_Rt3(Variables.Ri, Variables.Rr1, Variables.Rr2, Variables.Rex3);
System.out.println("A quantidade de Rt3 é: " + Variables.Rt3);
Resultados.resultados.append("Resistência térmica total da 3ª iteração (Rt3) é [m2/W K]:
").append(Variables.Rt3).append(System.getProperty("line.separator"));
Variables.q3 = equation7_q3(Variables.Tamb, Variables.Rt3);
System.out.println("A quantidade de q3 é: " + Variables.q3);
Resultados.resultados.append("O fluxo de calor da 3ª iteração (q3) é [kW/m2]:
").append(Variables.q3).append(System.getProperty("line.separator"));
Variables.Tpi3 = equation7_Tpi3(Variables.q3, Variables.Ri);
System.out.println("A quantidade de Tpi3 é: " + Variables.Tpi3);
Variables.Tr13 = equation7_Tr13(Variables.q3, Variables.Rr1, Variables.Tpi3);
System.out.println("A quantidade de Tr13 é: " + Variables.Tr13);
Variables.Tpex3 = equation7_Tpex3(Variables.q3, Variables.Rr2, Variables.Tr13);
System.out.println("A quantidade de Tpex3 é: " + Variables.Tpex3);
Resultados.resultados.append("A temperatura da parede exterior na 3ª iteração (Tpex3) é [°C]:
").append(Variables.Tpex3).append(System.getProperty("line.separator"));
Variables.Tpar4 = equation7_Tpar4(Variables.Tpex3);
System.out.println("A quantidade de Tpar4 é: " + Variables.Tpar4);
Variables.Kamb4 = equation7_Kamb4(Variables.Tf4);
System.out.println("A quantidade de Kamb4 é: " + Variables.Kamb4);

```

```
Resultados.resultados.append("Condutividade térmica do ar ambiente da 4ª iteração (Kamb4) é  
[W/mK]: ").append(Variables.Kamb4).append(System.getProperty("line.separator"));  
Variables.Tf4 = equation7_Tf4(Variables.Tpar4, Variables.Tamb);  
System.out.println("A quantidade de Tf4 é: " + Variables.Tf4);  
Resultados.resultados.append("A temperatura do filme da 4ª iteração (Tf4) é [°C]:  
").append(Variables.Tf4).append(System.getProperty("line.separator"));  
Variables.ViscoCin4 = equation7_ViscoCin4(Variables.Tf4);  
System.out.println("A quantidade de ViscoCin4 é: " + Variables.ViscoCin4);  
Resultados.resultados.append("A viscosidade cinemática do ar da 4ª iteração (ViscoCin4) é  
[m2/s]: ").append(Variables.ViscoCin4).append(System.getProperty("line.separator"));  
Variables.beta4 = equation7_beta4(Variables.Tf4);  
System.out.println("A quantidade de beta4 é: " + Variables.beta4);  
Resultados.resultados.append("Coeficiente de expansão volumétrica da 4ª iteração (beta4) é  
[1/K]: ").append(Variables.beta4).append(System.getProperty("line.separator"));  
Variables.Pr4 = equation7_Pr4(Variables.Tf4);  
System.out.println("A quantidade de Pr4 é: " + Variables.Pr4);  
Resultados.resultados.append("O número de Prandtl da 4ª iteração (Pr4) é:  
").append(Variables.Pr4).append(System.getProperty("line.separator"));  
Variables.Gr4 = equation7_Gr4(Variables.beta4, Variables.Tpar4, Variables.Tamb, Variables.Lc,  
Variables.ViscoCin4);  
System.out.println("A quantidade de Gr4 é: " + Variables.Gr4);  
Resultados.resultados.append("O número de Grashof da 4ª iteração (Gr4) é:  
").append(Variables.Gr4).append(System.getProperty("line.separator"));  
Variables.Ra4 = equation7_Ra4(Variables.Gr4, Variables.Pr4);  
System.out.println("A quantidade de Ra4 é: " + Variables.Ra4);  
Resultados.resultados.append("O número de Rayleigh da 4ª iteração (Ra4) é:  
").append(Variables.Ra4).append(System.getProperty("line.separator"));  
Variables.Nu4 = equation7_Nu4(Variables.Ra4, Variables.Pr4);  
System.out.println("A quantidade de Nu4 é: " + Variables.Nu4);  
Resultados.resultados.append("O número de Nusselt da 4ª iteração (Nu4) é:  
").append(Variables.Nu4).append(System.getProperty("line.separator"));  
Variables.hamb4 = equation7_hamb4(Variables.Nu4, Variables.Kamb4, Variables.Lc);  
System.out.println("A quantidade de hamb4 é: " + Variables.hamb4);  
Resultados.resultados.append("O coeficiente de convecção do ambiente da 4ª iteração (hamb4)  
é [W/m2K]: ").append(Variables.hamb4).append(System.getProperty("line.separator"));  
Variables.hef4 = equation7_hef4(Variables.hamb4, Variables.Tpar4, Variables.Tamb);  
System.out.println("A quantidade de hef4 é: " + Variables.hef4);
```

```

Resultados.resultados.append("O coeficiente de global (efetivo) da 4ª iteração (hef4) é [W/m2K]:
").append(Variables.hef4).append(System.getProperty("line.separator"));
Variables.Rex4 = equation7_Rex4(Variables.hef4, Variables.Lc);
System.out.println("A quantidade de Rex4 é: " + Variables.Rex4);
Variables.Rt4 = equation7_Rt4(Variables.Ri, Variables.Rr1, Variables.Rr2, Variables.Rex4);
System.out.println("A quantidade de Rt4 é: " + Variables.Rt4);
Resultados.resultados.append("Resistência térmica total da 4ª iteração (Rt4) é [m2/W K]:
").append(Variables.Rt4).append(System.getProperty("line.separator"));
Variables.q4 = equation7_q4(Variables.Tamb, Variables.Rt4);
System.out.println("A quantidade de q4 é: " + Variables.q4);
Resultados.resultados.append("O fluxo de calor da 4ª iteração (q4) é [kW/m2]:
").append(Variables.q4).append(System.getProperty("line.separator"));
Variables.Tpi4 = equation7_Tpi4(Variables.q4, Variables.Ri);
System.out.println("A quantidade de Tpi4 é: " + Variables.Tpi4);
Variables.Tr14 = equation7_Tr14(Variables.q4, Variables.Rr1, Variables.Tpi4);
System.out.println("A quantidade de Tr14 é: " + Variables.Tr14);
Variables.Tpex4 = equation7_Tpex4(Variables.q4, Variables.Rr2, Variables.Tr14);
System.out.println("A quantidade de Tpex4 é: " + Variables.Tpex4);
Resultados.resultados.append("A temperatura da parede exterior na 4ª iteração (Tpex4) é [°C]:
").append(Variables.Tpex4).append(System.getProperty("line.separator"));
Variables.Tpar5 = equation7_Tpar5(Variables.Tpex4);
System.out.println("A quantidade de Tpar5 é: " + Variables.Tpar5);
Variables.Tf5 = equation7_Tf5(Variables.Tpar5, Variables.Tamb);
System.out.println("A quantidade de Tf5 é: " + Variables.Tf5);
Resultados.resultados.append("A temperatura do filme da 5ª iteração (Tf5) é [°C]:
").append(Variables.Tf5).append(System.getProperty("line.separator"));
Variables.Kamb5 = equation7_Kamb5(Variables.Tf5);
System.out.println("A quantidade de Kamb5 é: " + Variables.Kamb5);
Resultados.resultados.append("Condutividade térmica do ar ambiente da 5ª iteração (Kamb5) é
[W/mK]: ").append(Variables.Kamb5).append(System.getProperty("line.separator"));
Variables.ViscoCin5 = equation7_ViscoCin5(Variables.Tf5);
System.out.println("A quantidade de ViscoCin5 é: " + Variables.ViscoCin5);
Resultados.resultados.append("A viscosidade cinemática do ar da 5ª iteração (ViscoCin5) é
[m2/s]: ").append(Variables.ViscoCin5).append(System.getProperty("line.separator"));
Variables.beta5 = equation7_beta5(Variables.Tf5);
System.out.println("A quantidade de beta5 é: " + Variables.beta5);
Resultados.resultados.append("Coeficiente de expansão volumétrica da 5ª iteração (beta5) é
[1/K]: ").append(Variables.beta5).append(System.getProperty("line.separator"));

```

```

Variables.Pr5 = equation7_Pr5(Variables.Tf5);
System.out.println("A quantidade de Pr5 é: " + Variables.Pr5);
Resultados.resultados.append("O número de Prandtl da 5ª iteração (Pr5) é:
").append(Variables.Pr5).append(System.getProperty("line.separator"));
Variables.Gr5 = equation7_Gr5(Variables.beta5, Variables.Tpar5, Variables.Tamb, Variables.Lc,
Variables.ViscoCin5);
System.out.println("A quantidade de Gr5 é: " + Variables.Gr5);
Resultados.resultados.append("O número de Grashof da 5ª iteração (Gr5) é:
").append(Variables.Gr5).append(System.getProperty("line.separator"));
Variables.Ra5 = equation7_Ra5(Variables.Gr5, Variables.Pr5);
System.out.println("A quantidade de Ra5 é: " + Variables.Ra5);
Resultados.resultados.append("O número de Rayleigh da 5ª iteração (Ra5) é:
").append(Variables.Ra5).append(System.getProperty("line.separator"));
Variables.Nu5 = equation7_Nu5(Variables.Ra5, Variables.Pr5);
System.out.println("A quantidade de Nu5 é: " + Variables.Nu5);
Resultados.resultados.append("O número de Nusselt da 5ª iteração (Nu5) é:
").append(Variables.Nu5).append(System.getProperty("line.separator"));
Variables.hamb5 = equation7_hamb5(Variables.Nu5, Variables.Kamb5, Variables.Lc);
System.out.println("A quantidade de hamb5 é: " + Variables.hamb5);
Resultados.resultados.append("O coeficiente de convecção do ambiente da 5ª iteração (hamb5)
é [W/m2K]: ").append(Variables.hamb5).append(System.getProperty("line.separator"));
Variables.hef5 = equation7_hef5(Variables.hamb5, Variables.Tpar5, Variables.Tamb);
System.out.println("A quantidade de hef5 é: " + Variables.hef5);
Resultados.resultados.append("O coeficiente de global (efetivo) da 5ª iteração (hef5) é [W/m2K]:
").append(Variables.hef5).append(System.getProperty("line.separator"));
Variables.Rex5 = equation7_Rex5(Variables.hef5, Variables.Lc);
System.out.println("A quantidade de Rex5 é: " + Variables.Rex5);
Variables.Rt5 = equation7_Rt(Variables.Ri, Variables.Rr1, Variables.Rr2, Variables.Rex5);
System.out.println("A quantidade de Rt5 é: " + Variables.Rt5);
Resultados.resultados.append("Resistência térmica total da 5ª iteração (Rt5) é [m2/W K]:
").append(Variables.Rt5).append(System.getProperty("line.separator"));
Variables.q5 = equation7_q5(Variables.Tamb, Variables.Rt5);
System.out.println("A quantidade de q5 é: " + Variables.q5);
Resultados.resultados.append("O fluxo de calor da 5ª iteração (q5) é [kW/m2]:
").append(Variables.q5).append(System.getProperty("line.separator"));
Variables.Tpi5 = equation7_Tpi5(Variables.q5, Variables.Ri);
System.out.println("A quantidade de Tpi5 é: " + Variables.Tpi5);
Variables.Tr15 = equation7_Tr15(Variables.q5, Variables.Rr1, Variables.Tpi5);

```



```

System.out.println("A quantidade de Tr1 é: " + Variables.Tr15);
Variables.Tpex5 = equation7_Tpex5(Variables.q5, Variables.Rr2, Variables.Tr15);
System.out.println("A quantidade de Tpex5 é: " + Variables.Tpex5);
Resultados.resultados.append("A temperatura da parede exterior na 5ª iteração (Tpex5) é [°C]:
").append(Variables.Tpex5).append(System.getProperty("line.separator"));
Variables.Tpar = equation7_Tpar(Variables.Tpex5);
System.out.println("A quantidade de Tpar é: " + Variables.Tpar);
Variables.Tf = equation7_Tf(Variables.Tpar, Variables.Tamb);
System.out.println("A quantidade de Tf é: " + Variables.Tf);
Resultados.resultados.append("A temperatura do filme (6ª iteração) (Tf) é [°C]:
").append(Variables.Tf).append(System.getProperty("line.separator"));
Variables.Kamb = equation7_Kamb(Variables.Tf);
System.out.println("A quantidade de Kamb é: " + Variables.Kamb);
Resultados.resultados.append("Condutividade térmica do ar ambiente (6ª iteração) (Kamb) é
[W/mK]: ").append(Variables.Kamb).append(System.getProperty("line.separator"));
Variables.ViscoCin = equation7_ViscoCin(Variables.Tf);
System.out.println("A quantidade de ViscoCin é: " + Variables.ViscoCin);
Resultados.resultados.append("A viscosidade cinemática do ar (6ª iteração) (ViscoCin1) é [m2/s]:
").append(Variables.ViscoCin).append(System.getProperty("line.separator"));
Variables.beta = equation7_beta(Variables.Tf);
System.out.println("A quantidade de beta é: " + Variables.beta);
Resultados.resultados.append("Coeficiente de expansão volumétrica (6ª iteração) (beta) é [1/K]:
").append(Variables.beta).append(System.getProperty("line.separator"));
Variables.Pr = equation7_Pr(Variables.Tf);
System.out.println("A quantidade de Pr é: " + Variables.Pr);
Resultados.resultados.append("O número de Prandtl (6ª iteração) (Pr) é:
").append(Variables.Pr).append(System.getProperty("line.separator"));
Variables.Gr = equation7_Gr(Variables.beta, Variables.Tpar, Variables.Tamb, Variables.Lc,
Variables.ViscoCin);
System.out.println("A quantidade de Gr é: " + Variables.Gr);
Resultados.resultados.append("O número de Grashof (6ª iteração) (Gr) é:
").append(Variables.Gr).append(System.getProperty("line.separator"));
Variables.Ra = equation7_Ra(Variables.Gr, Variables.Pr);
System.out.println("A quantidade de Ra é: " + Variables.Ra);
Resultados.resultados.append("O número de Rayleigh da (6ª iteração) (Ra) é:
").append(Variables.Ra).append(System.getProperty("line.separator"));
Variables.Nu = equation7_Nu(Variables.Ra, Variables.Pr);
System.out.println("A quantidade de Nu é: " + Variables.Nu);

```

```

Resultados.resultados.append("O número de Nusselt (6ª iteração) (Nu) é:
").append(Variables.Nu).append(System.getProperty("line.separator"));
Variables.hamb = equation7_hamb(Variables.Nu, Variables.Kamb, Variables.Lc);
System.out.println("A quantidade de hamb é: " + Variables.hamb);
Resultados.resultados.append("O coeficiente de convecção do ambiente (6ª iteração) (hamb) é
[W/m2K]: ").append(Variables.hamb).append(System.getProperty("line.separator"));
Variables.hef = equation7_hef(Variables.hamb, Variables.Tpar, Variables.Tamb);
System.out.println("A quantidade de hef é: " + Variables.hef);
Resultados.resultados.append("O coeficiente de global (efetivo) (6ª iteração) (hef) é [W/m2K]:
").append(Variables.hef).append(System.getProperty("line.separator"));
Variables.Rex = equation7_Rex(Variables.hef, Variables.Lc);
System.out.println("A quantidade de Rex é: " + Variables.Rex);
Variables.Rt = equation7_Rt(Variables.Ri, Variables.Rr1, Variables.Rr2, Variables.Rex);
System.out.println("A quantidade de Rt é: " + Variables.Rt);
Resultados.resultados.append("Resistência térmica total (6ª iteração) (Rt) é [m2/W K]:
").append(Variables.Rt).append(System.getProperty("line.separator"));
Variables.q = equation7_q(Variables.Tamb, Variables.Rt);
System.out.println("A quantidade de q é: " + Variables.q);
Resultados.resultados.append("O fluxo de calor (6ª iteração) (q) é [kW/m2]:
").append(Variables.q).append(System.getProperty("line.separator"));
Variables.Tpi = equation7_Tpi(Variables.q, Variables.Ri);
System.out.println("A quantidade de Tpi é: " + Variables.Tpi);
Variables.Tr1 = equation7_Tr1(Variables.q, Variables.Rr1, Variables.Tpi);
System.out.println("A quantidade de Tr1 é: " + Variables.Tr1);
Variables.Tpex = equation7_Tpex(Variables.q, Variables.Rr2, Variables.Tr1);
System.out.println("A temperatura da parede exterior na zona mais crítica (zona de calcinação) é
(Tpex): " + Variables.Tpex);
Resultados.resultados.append("A temperatura da parede exterior na zona mais crítica (6ª
iteração) (Tpex) é [°C]: ").append(Variables.Tpex).append(System.getProperty("line.separator"));
if (Variables.Df <= Variables.D){
    Variables.Df = equation7_Df(Variables.Lc, Variables.Gr1);
    Resultados.resultados.append("----- Fim da Fase 7 COM SUCESSO -----
").append(System.getProperty("line.separator")).append(System.getProperty("line.separator"));
    return true;
}
else{
    System.out.println("ERROR");

```

```
Resultados.resultados.append("----- Fim da Fase 7 SEM SUCESSO: Df não é menor que  
D -----  
").append(System.getProperty("line.separator")).append(System.getProperty("line.separator"));  
    return false;  
}  
}  
public static double equation7_r1(double R){  
    double r1 = 0.0;  
    r1 = Fase1.Variables.R + Constants.Lr1;  
    return r1;  
}  
public static double equation7_r2(double r1){  
    double r2 = 0.0;  
    r2 = r1+Constants.Lr2;  
    return r2;  
}  
public static double equation7_Lc(double H){  
    double Lc = 0.0;  
    Lc = Fase1.Variables.H;  
    return Lc;  
}  
public static double equation7_D(double r2){  
    double D = 0.0;  
    D = r2*2;  
    return D;  
}  
public static double equation7_esp(double D, double R){  
    double esp = 0.0;  
    esp = (D/2)-R;  
    return esp;  
}  
public static double equation7_Re2(double dp){  
    double Re2 = 0.0;  
    Re2 = Fase1.Variables.dp*0.001*Constants.Uf2*Constants.rhoG2/Fase4.Constants.mu;  
    return Re2;  
}  
public static double equation7_hi(double Re2){  
    double hi = 0.0;
```



```
hi = 0.525*(Math.pow(Re2,0.75));
return hi;
}
public static double equation7_Tpar2(double Tpex1){
    double Tpar2 = 0.0;
    Tpar2 = Tpex1;
    return Tpar2;
}
public static double equation7_Tpar3(double Tpex2){
    double Tpar3 = 0.0;
    Tpar3 = Tpex2;
    return Tpar3;
}
public static double equation7_Tpar4(double Tpex3){
    double Tpar4 = 0.0;
    Tpar4 = Tpex3;
    return Tpar4;
}
public static double equation7_Tpar5(double Tpex4){
    double Tpar5 = 0.0;
    Tpar5 = Tpex4;
    return Tpar5;
}
public static double equation7_Tpar(double Tpex5){
    double Tpar = 0.0;
    Tpar = Tpex5;
    return Tpar;
}
public static double equation7_Tf1(double Tamb){
    double Tf1 = 0.0;
    Tf1 = (Constants.Tpar1+Tamb)/2;
    return Tf1;
}
public static double equation7_Tf2(double Tpar2, double Tamb){
    double Tf2 = 0.0;
    Tf2 = (Tpar2+Tamb)/2;
    return Tf2;
}
```

```

public static double equation7_Tf3(double Tpar3, double Tamb){
    double Tf3 = 0.0;
    Tf3 = (Tpar3+Tamb)/2;
    return Tf3;
}

public static double equation7_Tf4(double Tpar4, double Tamb){
    double Tf4 = 0.0;
    Tf4 = (Tpar4+Tamb)/2;
    return Tf4;
}

public static double equation7_Tf5(double Tpar5, double Tamb){
    double Tf5 = 0.0;
    Tf5 = (Tpar5+Tamb)/2;
    return Tf5;
}

public static double equation7_Tf(double Tpar, double Tamb){
    double Tf = 0.0;
    Tf = (Tpar+Tamb)/2;
    return Tf;
}

public static double equation7_Kamb1(double Tf1){
    double Kamb1 = 0.0;
    Kamb1 = -1.1198897E-20*Math.pow(Tf1,6)+8.7864215E-15*Math.pow(Tf1,5)-4.4374479E-
14*Math.pow(Tf1,4)+9.6559189E-10*Math.pow(Tf1,3)-6.7839993E-8*Math.pow(Tf1,2)+7.1477994E-
5*Tf1+2.4295037E-2;
    return Kamb1;
}

public static double equation7_Kamb2(double Tf2){
    double Kamb2 = 0.0;
    Kamb2 = -1.1198897E-20*Math.pow(Tf2,6)+8.7864215E-15*Math.pow(Tf2,5)-4.4374479E-
14*Math.pow(Tf2,4)+9.6559189E-10*Math.pow(Tf2,3)-6.7839993E-8*Math.pow(Tf2,2)+7.1477994E-
5*Tf2+2.4295037E-2;
    return Kamb2;
}

public static double equation7_Kamb3(double Tf3){
    double Kamb3 = 0.0;

```

```

    Kamb3 = -1.1198897E-20*Math.pow(Tf3,6)+8.7864215E-15*Math.pow(Tf3,5)-4.4374479E-
14*Math.pow(Tf3,4)+9.6559189E-10*Math.pow(Tf3,3)-6.7839993E-8*Math.pow(Tf3,2)+7.1477994E-
5*Tf3+2.4295037E-2;
    return Kamb3;
}

public static double equation7_Kamb4(double Tf4){
    double Kamb4 = 0.0;
    Kamb4 = -1.1198897E-20*Math.pow(Tf4,6)+8.7864215E-15*Math.pow(Tf4,5)-4.4374479E-
14*Math.pow(Tf4,4)+9.6559189E-10*Math.pow(Tf4,3)-6.7839993E-8*Math.pow(Tf4,2)+7.1477994E-
5*Tf4+2.4295037E-2;
    return Kamb4;
}

public static double equation7_Kamb5(double Tf5){
    double Kamb5 = 0.0;
    Kamb5 = -1.1198897E-20*Math.pow(Tf5,6)+8.7864215E-15*Math.pow(Tf5,5)-4.4374479E-
14*Math.pow(Tf5,4)+9.6559189E-10*Math.pow(Tf5,3)-6.7839993E-8*Math.pow(Tf5,2)+7.1477994E-
5*Tf5+2.4295037E-2;
    return Kamb5;
}

public static double equation7_Kamb(double Tf){
    double Kamb = 0.0;
    Kamb = -1.1198897E-20*Math.pow(Tf,6)+8.7864215E-15*Math.pow(Tf,5)-4.4374479E-
14*Math.pow(Tf,4)+9.6559189E-10*Math.pow(Tf,3)-6.7839993E-8*Math.pow(Tf,2)+7.1477994E-
5*Tf+2.4295037E-2;
    return Kamb;
}

public static double equation7_ViscoCin1(double Tf1){
    double ViscoCin1 = 0.0;
    ViscoCin1 = 4.365140866895E-20*Math.pow(Tf1,6)-4.039290616056E-
17*Math.pow(Tf1,5)+1.088015085567E-14*Math.pow(Tf1,4)-1.141636784366E-
12*Math.pow(Tf1,3)+1.444436360192E-10*Math.pow(Tf1,2)+8.699748434118E-
8*Tf1+1.331422047548E-5;
    return ViscoCin1;
}

public static double equation7_ViscoCin2(double Tf2){
    double ViscoCin2 = 0.0;
    ViscoCin2 = 4.365140866895E-20*Math.pow(Tf2,6)-4.039290616056E-
17*Math.pow(Tf2,5)+1.088015085567E-14*Math.pow(Tf2,4)-1.141636784366E-

```

```

12*Math.pow(Tf2,3)+1.444436360192E-10*Math.pow(Tf2,2)+8.699748434118E-
8*Tf2+1.331422047548E-5;
    return ViscoCin2;
}

public static double equation7_ViscoCin3(double Tf3){
    double ViscoCin3 = 0.0;
    ViscoCin3 = 4.365140866895E-20*Math.pow(Tf3,6)-4.039290616056E-
17*Math.pow(Tf3,5)+1.088015085567E-14*Math.pow(Tf3,4)-1.141636784366E-
12*Math.pow(Tf3,3)+1.444436360192E-10*Math.pow(Tf3,2)+8.699748434118E-
8*Tf3+1.331422047548E-5;
    return ViscoCin3;
}

public static double equation7_ViscoCin4(double Tf4){
    double ViscoCin4 = 0.0;
    ViscoCin4 = 4.365140866895E-20*Math.pow(Tf4,6)-4.039290616056E-
17*Math.pow(Tf4,5)+1.088015085567E-14*Math.pow(Tf4,4)-1.141636784366E-
12*Math.pow(Tf4,3)+1.444436360192E-10*Math.pow(Tf4,2)+8.699748434118E-
8*Tf4+1.331422047548E-5;
    return ViscoCin4;
}

public static double equation7_ViscoCin5(double Tf5){
    double ViscoCin5 = 0.0;
    ViscoCin5 = 4.365140866895E-20*Math.pow(Tf5,6)-4.039290616056E-
17*Math.pow(Tf5,5)+1.088015085567E-14*Math.pow(Tf5,4)-1.141636784366E-
12*Math.pow(Tf5,3)+1.444436360192E-10*Math.pow(Tf5,2)+8.699748434118E-
8*Tf5+1.331422047548E-5;
    return ViscoCin5;
}

public static double equation7_ViscoCin(double Tf){
    double ViscoCin = 0.0;
    ViscoCin = 4.365140866895E-20*Math.pow(Tf,6)-4.039290616056E-
17*Math.pow(Tf,5)+1.088015085567E-14*Math.pow(Tf,4)-1.141636784366E-
12*Math.pow(Tf,3)+1.444436360192E-10*Math.pow(Tf,2)+8.699748434118E-
8*Tf+1.331422047548E-5;
    return ViscoCin;
}

public static double equation7_beta1(double Tf1){
    double beta1 = 0.0;

```

```
beta1 = 5.8070940E-17*Math.pow(Tf1,6)-2.9994199E-14*Math.pow(Tf1,5)+5.1941928E-
12*Math.pow(Tf1,4)-3.7374860E-10*Math.pow(Tf1,3)+3.9708796E-8*Math.pow(Tf1,2)-1.2953005E-
5*Tf1+3.6734906E-3;
return beta1;
}
public static double equation7_beta2(double Tf2){
double beta2 = 0.0;
beta2 = 5.8070940E-17*Math.pow(Tf2,6)-2.9994199E-14*Math.pow(Tf2,5)+5.1941928E-
12*Math.pow(Tf2,4)-3.7374860E-10*Math.pow(Tf2,3)+3.9708796E-8*Math.pow(Tf2,2)-1.2953005E-
5*Tf2+3.6734906E-3;
return beta2;
}
public static double equation7_beta3(double Tf3){
double beta3 = 0.0;
beta3 = 5.8070940E-17*Math.pow(Tf3,6)-2.9994199E-14*Math.pow(Tf3,5)+5.1941928E-
12*Math.pow(Tf3,4)-3.7374860E-10*Math.pow(Tf3,3)+3.9708796E-8*Math.pow(Tf3,2)-1.2953005E-
5*Tf3+3.6734906E-3;
return beta3;
}
public static double equation7_beta4(double Tf4){
double beta4 = 0.0;
beta4 = 5.8070940E-17*Math.pow(Tf4,6)-2.9994199E-14*Math.pow(Tf4,5)+5.1941928E-
12*Math.pow(Tf4,4)-3.7374860E-10*Math.pow(Tf4,3)+3.9708796E-8*Math.pow(Tf4,2)-1.2953005E-
5*Tf4+3.6734906E-3;
return beta4;
}
public static double equation7_beta5(double Tf5){
double beta5 = 0.0;
beta5 = 5.8070940E-17*Math.pow(Tf5,6)-2.9994199E-14*Math.pow(Tf5,5)+5.1941928E-
12*Math.pow(Tf5,4)-3.7374860E-10*Math.pow(Tf5,3)+3.9708796E-8*Math.pow(Tf5,2)-1.2953005E-
5*Tf5+3.6734906E-3;
return beta5;
}
public static double equation7_beta(double Tf){
double beta = 0.0;
beta = 5.8070940E-17*Math.pow(Tf,6)-2.9994199E-14*Math.pow(Tf,5)+5.1941928E-
12*Math.pow(Tf,4)-3.7374860E-10*Math.pow(Tf,3)+3.9708796E-8*Math.pow(Tf,2)-1.2953005E-
5*Tf+3.6734906E-3;
```

```

    return beta;
}

public static double equation7_Pr1(double Tf1){
    double Pr1 = 0.0;

    Pr1 = -8.9663606E-15*Math.pow(Tf1,6)+4.2115168E-12*Math.pow(Tf1,5)-5.8250457E-
10*Math.pow(Tf1,4)+8.6537272E-9*Math.pow(Tf1,3)+2.4053808E-6*Math.pow(Tf1,2)-1.9350745E-
4*Tf1+7.1547018E-1;

    return Pr1;
}

public static double equation7_Pr2(double Tf2){
    double Pr2 = 0.0;

    Pr2 = -8.9663606E-15*Math.pow(Tf2,6)+4.2115168E-12*Math.pow(Tf2,5)-5.8250457E-
10*Math.pow(Tf2,4)+8.6537272E-9*Math.pow(Tf2,3)+2.4053808E-6*Math.pow(Tf2,2)-1.9350745E-
4*Tf2+7.1547018E-1;

    return Pr2;
}

public static double equation7_Pr3(double Tf3){
    double Pr3 = 0.0;

    Pr3 = -8.9663606E-15*Math.pow(Tf3,6)+4.2115168E-12*Math.pow(Tf3,5)-5.8250457E-
10*Math.pow(Tf3,4)+8.6537272E-9*Math.pow(Tf3,3)+2.4053808E-6*Math.pow(Tf3,2)-1.9350745E-
4*Tf3+7.1547018E-1;

    return Pr3;
}

public static double equation7_Pr4(double Tf4){
    double Pr4 = 0.0;

    Pr4 = -8.9663606E-15*Math.pow(Tf4,6)+4.2115168E-12*Math.pow(Tf4,5)-5.8250457E-
10*Math.pow(Tf4,4)+8.6537272E-9*Math.pow(Tf4,3)+2.4053808E-6*Math.pow(Tf4,2)-1.9350745E-
4*Tf4+7.1547018E-1;

    return Pr4;
}

public static double equation7_Pr5(double Tf5){
    double Pr5 = 0.0;

    Pr5 = -8.9663606E-15*Math.pow(Tf5,6)+4.2115168E-12*Math.pow(Tf5,5)-5.8250457E-
10*Math.pow(Tf5,4)+8.6537272E-9*Math.pow(Tf5,3)+2.4053808E-6*Math.pow(Tf5,2)-1.9350745E-
4*Tf5+7.1547018E-1;

    return Pr5;
}

public static double equation7_Pr(double Tf){

```

```

double Pr = 0.0;
Pr = -8.9663606E-15*Math.pow(Tf,6)+4.2115168E-12*Math.pow(Tf,5)-5.8250457E-
10*Math.pow(Tf,4)+8.6537272E-9*Math.pow(Tf,3)+2.4053808E-6*Math.pow(Tf,2)-1.9350745E-
4*Tf+7.1547018E-1;
return Pr;
}
public static double equation7_Gr1(double beta1, double Tamb, double Lc, double ViscoCin1){
double Gr1 = 0.0;
Gr1 = (Constants.g*beta1*(Constants.Tpar1-Tamb)*(Math.pow(Lc, 3)))/(Math.pow(ViscoCin1, 2));
return Gr1;
}
public static double equation7_Df(double Lc, double Gr1){
double Df = 0.0;
Df = 35*Lc/(Math.pow(Gr1,0.25));
return Df;
}
public static double equation7_Gr2(double beta2, double Tpar2, double Tamb, double Lc, double
ViscoCin2){
double Gr2 = 0.0;
Gr2 = (Constants.g*beta2*(Tpar2-Tamb)*(Math.pow(Lc, 3)))/(Math.pow(ViscoCin2, 2));
return Gr2;
}
public static double equation7_Gr3(double beta3, double Tpar3, double Tamb, double Lc, double
ViscoCin3){
double Gr3 = 0.0;
Gr3 = (Constants.g*beta3*(Tpar3-Tamb)*(Math.pow(Lc, 3)))/(Math.pow(ViscoCin3, 2));
return Gr3;
}
public static double equation7_Gr4(double beta4, double Tpar4, double Tamb, double Lc, double
ViscoCin4){
double Gr4 = 0.0;
Gr4 = (Constants.g*beta4*(Tpar4-Tamb)*(Math.pow(Lc, 3)))/(Math.pow(ViscoCin4, 2));
return Gr4;
}
public static double equation7_Gr5(double beta5, double Tpar5, double Tamb, double Lc, double
ViscoCin5){
double Gr5 = 0.0;
Gr5 = (Constants.g*beta5*(Tpar5-Tamb)*(Math.pow(Lc, 3)))/(Math.pow(ViscoCin5, 2));

```

```
    return Gr5;
}

public static double equation7_Gr(double beta, double Tpar, double Tamb, double Lc, double
ViscoCin){
    double Gr = 0.0;
    Gr = (Constants.g*beta*(Tpar-Tamb)*(Math.pow(Lc, 3)))/(Math.pow(ViscoCin, 2));
    return Gr;
}

public static double equation7_Ra1(double Gr1, double Pr1){
    double Ra1 = 0.0;
    Ra1 = Gr1*Pr1;
    return Ra1;
}

public static double equation7_Ra2(double Gr2, double Pr2){
    double Ra2 = 0.0;
    Ra2 = Gr2*Pr2;
    return Ra2;
}

public static double equation7_Ra3(double Gr3, double Pr3){
    double Ra3 = 0.0;
    Ra3 = Gr3*Pr3;
    return Ra3;
}

public static double equation7_Ra4(double Gr4, double Pr4){
    double Ra4 = 0.0;
    Ra4 = Gr4*Pr4;
    return Ra4;
}

public static double equation7_Ra5(double Gr5, double Pr5){
    double Ra5 = 0.0;
    Ra5 = Gr5*Pr5;
    return Ra5;
}

public static double equation7_Ra(double Gr, double Pr){
    double Ra = 0.0;
    Ra = Gr*Pr;
    return Ra;
}
```



```

public static double equation7_Nu(double Ra, double Pr){
    double Nu = 0.0;
    Nu =
    Math.pow(0.825+0.387*(Math.pow(Ra,0.166666667))/((Math.pow(1+(Math.pow(0.492/Pr,0.5625))),0.29
    6296296))),2);
    return Nu;
}

public static double equation7_Nu1(double Ra1, double Pr1){
    double Nu1 = 0.0;
    Nu1 =
    Math.pow(0.825+0.387*(Math.pow(Ra1,0.166666667))/((Math.pow(1+(Math.pow(0.492/Pr1,0.5625))),0.
    296296296))),2);
    return Nu1;
}

public static double equation7_Nu2(double Ra2, double Pr2){
    double Nu2 = 0.0;
    Nu2 =
    Math.pow(0.825+0.387*(Math.pow(Ra2,0.166666667))/((Math.pow(1+(Math.pow(0.492/Pr2,0.5625))),0.
    296296296))),2);
    return Nu2;
}

public static double equation7_Nu3(double Ra3, double Pr3){
    double Nu3 = 0.0;
    Nu3 =
    Math.pow(0.825+0.387*(Math.pow(Ra3,0.166666667))/((Math.pow(1+(Math.pow(0.492/Pr3,0.5625))),0.
    296296296))),2);
    return Nu3;
}

public static double equation7_Nu4(double Ra4, double Pr4){
    double Nu4 = 0.0;
    Nu4 =
    Math.pow(0.825+0.387*(Math.pow(Ra4,0.166666667))/((Math.pow(1+(Math.pow(0.492/Pr4,0.5625))),0.
    296296296))),2);
    return Nu4;
}

public static double equation7_Nu5(double Ra5, double Pr5){
    double Nu5 = 0.0;

```

```

    Nu5 =
    Math.pow(0.825+0.387*(Math.pow(Ra5,0.166666667))/((Math.pow(1+(Math.pow(0.492/Pr5,0.5625))),0.
    296296296))),2);
    return Nu5;
}
public static double equation7_hamb1(double Nu1, double Kamb1, double Lc){
    double hamb1 = 0.0;
    hamb1 = Nu1*Kamb1/Lc;
    return hamb1;
}
public static double equation7_hamb2(double Nu2, double Kamb2, double Lc){
    double hamb2 = 0.0;
    hamb2 = Nu2*Kamb2/Lc;
    return hamb2;
}
public static double equation7_hamb3(double Nu3, double Kamb3, double Lc){
    double hamb3 = 0.0;
    hamb3 = Nu3*Kamb3/Lc;
    return hamb3;
}
public static double equation7_hamb4(double Nu4, double Kamb4, double Lc){
    double hamb4 = 0.0;
    hamb4 = Nu4*Kamb4/Lc;
    return hamb4;
}
public static double equation7_hamb5(double Nu5, double Kamb5, double Lc){
    double hamb5 = 0.0;
    hamb5 = Nu5*Kamb5/Lc;
    return hamb5;
}
public static double equation7_hamb(double Nu, double Kamb, double Lc){
    double hamb = 0.0;
    hamb = Nu*Kamb/Lc;
    return hamb;
}
public static double equation7_hef1(double hamb1, double Tamb){
    double hef1 = 0.0;

```

```
    hef1 = (hamb1*(Constants.Tpar1-
Tamb)+1*Constants.Emi*Constants.sigma*((Math.pow(Constants.Tpar1+273.15,4))-
(Math.pow(Tamb+273.15,4))))/(Constants.Tpar1-Tamb);
    return hef1;
}

public static double equation7_hef2(double hamb2, double Tpar2, double Tamb){
    double hef2 = 0.0;
    hef2 = (hamb2*(Tpar2-Tamb)+1*Constants.Emi*Constants.sigma*((Math.pow(Tpar2+273.15,4))-
(Math.pow(Tamb+273.15,4))))/(Tpar2-Tamb);
    return hef2;
}

public static double equation7_hef3(double hamb3, double Tpar3, double Tamb){
    double hef3 = 0.0;
    hef3 = (hamb3*(Tpar3-Tamb)+1*Constants.Emi*Constants.sigma*((Math.pow(Tpar3+273.15,4))-
(Math.pow(Tamb+273.15,4))))/(Tpar3-Tamb);
    return hef3;
}

public static double equation7_hef4(double hamb4, double Tpar4, double Tamb){
    double hef4 = 0.0;
    hef4 = (hamb4*(Tpar4-Tamb)+1*Constants.Emi*Constants.sigma*((Math.pow(Tpar4+273.15,4))-
(Math.pow(Tamb+273.15,4))))/(Tpar4-Tamb);
    return hef4;
}

public static double equation7_hef5(double hamb5, double Tpar5, double Tamb){
    double hef5 = 0.0;
    hef5 = (hamb5*(Tpar5-Tamb)+1*Constants.Emi*Constants.sigma*((Math.pow(Tpar5+273.15,4))-
(Math.pow(Tamb+273.15,4))))/(Tpar5-Tamb);
    return hef5;
}

public static double equation7_hef(double hamb, double Tpar, double Tamb){
    double hef = 0.0;
    hef = (hamb*(Tpar-Tamb)+1*Constants.Emi*Constants.sigma*((Math.pow(Tpar+273.15,4))-
(Math.pow(Tamb+273.15,4))))/(Tpar-Tamb);
    return hef;
}

public static double equation7_Ri(double hi, double Lc){
    double Ri = 0.0;
    Ri = 1/(2*Math.PI*hi*Lc);
```

```
    return Ri;
}

public static double equation7_Rr1(double r1, double R, double Lc){
    double Rr1 = 0.0;
    Rr1 = (Math.log(r1/Fase1.Variables.R))/(2*Math.PI*Constants.Kr1*Lc);
    return Rr1;
}

public static double equation7_Rr2(double r1, double r2, double Lc){
    double Rr2 = 0.0;
    Rr2 = (Math.log(r2/r1))/(2*Math.PI*Constants.Kr2*Lc);
    return Rr2;
}

public static double equation7_Rex1(double hef1, double Lc){
    double Rex1 = 0.0;
    Rex1 = 1/(2*Math.PI*hef1*Lc);
    return Rex1;
}

public static double equation7_Rex2(double hef2, double Lc){
    double Rex2 = 0.0;
    Rex2 = 1/(2*Math.PI*hef2*Lc);
    return Rex2;
}

public static double equation7_Rex3(double hef3, double Lc){
    double Rex3 = 0.0;
    Rex3 = 1/(2*Math.PI*hef3*Lc);
    return Rex3;
}

public static double equation7_Rex4(double hef4, double Lc){
    double Rex4 = 0.0;
    Rex4 = 1/(2*Math.PI*hef4*Lc);
    return Rex4;
}

public static double equation7_Rex5(double hef5, double Lc){
    double Rex5 = 0.0;
    Rex5 = 1/(2*Math.PI*hef5*Lc);
    return Rex5;
}

public static double equation7_Rex(double hef, double Lc){
```

```
double Rex = 0.0;
Rex = 1/(2*Math.PI*hef*Lc);
return Rex;
}
public static double equation7_Rt1(double Ri, double Rr1, double Rr2, double Rex1){
double Rt1 = 0.0;
Rt1 = Ri+Rr1+Rr2+Rex1;
return Rt1;
}
public static double equation7_Rt2(double Ri, double Rr1, double Rr2, double Rex2){
double Rt2 = 0.0;
Rt2 = Ri+Rr1+Rr2+Rex2;
return Rt2;
}
public static double equation7_Rt3(double Ri, double Rr1, double Rr2, double Rex3){
double Rt3 = 0.0;
Rt3 = Ri+Rr1+Rr2+Rex3;
return Rt3;
}
public static double equation7_Rt4(double Ri, double Rr1, double Rr2, double Rex4){
double Rt4 = 0.0;
Rt4 = Ri+Rr1+Rr2+Rex4;
return Rt4;
}
public static double equation7_Rt5(double Ri, double Rr1, double Rr2, double Rex5){
double Rt5 = 0.0;
Rt5 = Ri+Rr1+Rr2+Rex5;
return Rt5;
}
public static double equation7_Rt(double Ri, double Rr1, double Rr2, double Rex){
double Rt = 0.0;
Rt = Ri+Rr1+Rr2+Rex;
return Rt;
}
public static double equation7_q1(double Tamb, double Rt1){
double q1 = 0.0;
q1 = (Constants.Ti-Tamb)/Rt1;
return q1;
}
```

```
}  
public static double equation7_q2(double Tamb, double Rt2){  
    double q2 = 0.0;  
    q2 = (Constants.Ti-Tamb)/Rt2;  
    return q2;  
}  
public static double equation7_q3(double Tamb, double Rt3){  
    double q3 = 0.0;  
    q3 = (Constants.Ti-Tamb)/Rt3;  
    return q3;  
}  
public static double equation7_q4(double Tamb, double Rt4){  
    double q4 = 0.0;  
    q4 = (Constants.Ti-Tamb)/Rt4;  
    return q4;  
}  
public static double equation7_q5(double Tamb, double Rt5){  
    double q5 = 0.0;  
    q5 = (Constants.Ti-Tamb)/Rt5;  
    return q5;  
}  
public static double equation7_q(double Tamb, double Rt){  
    double q = 0.0;  
    q = (Constants.Ti-Tamb)/Rt;  
    return q;  
}  
public static double equation7_Tpi1(double q1, double Ri){  
    double Tpi1 = 0.0;  
    Tpi1 = - (q1*Ri-Constants.Ti);  
    return Tpi1;  
}  
public static double equation7_Tpi2(double q2, double Ri){  
    double Tpi2 = 0.0;  
    Tpi2 = - (q2*Ri-Constants.Ti);  
    return Tpi2;  
}  
public static double equation7_Tpi3(double q3, double Ri){  
    double Tpi3 = 0.0;
```

```
Tpi3 = - (q3*Ri-Constants.Ti);
return Tpi3;
}
public static double equation7_Tpi4(double q4, double Ri){
    double Tpi4 = 0.0;
    Tpi4 = - (q4*Ri-Constants.Ti);
    return Tpi4;
}
public static double equation7_Tpi5(double q5, double Ri){
    double Tpi5 = 0.0;
    Tpi5 = - (q5*Ri-Constants.Ti);
    return Tpi5;
}
public static double equation7_Tpi(double q, double Ri){
    double Tpi = 0.0;
    Tpi = - (q*Ri-Constants.Ti);
    return Tpi;
}
public static double equation7_Tr11(double q1, double Rr1, double Tpi1){
    double Tr11 = 0.0;
    Tr11 = -(q1*Rr1-Tpi1);
    return Tr11;
}
public static double equation7_Tr12(double q2, double Rr1, double Tpi2){
    double Tr12 = 0.0;
    Tr12 = -(q2*Rr1-Tpi2);
    return Tr12;
}
public static double equation7_Tr13(double q3, double Rr1, double Tpi3){
    double Tr13 = 0.0;
    Tr13 = -(q3*Rr1-Tpi3);
    return Tr13;
}
public static double equation7_Tr14(double q4, double Rr1, double Tpi4){
    double Tr14 = 0.0;
    Tr14 = -(q4*Rr1-Tpi4);
    return Tr14;
}
```

```
public static double equation7_Tr15(double q5, double Rr1, double Tpi5){
    double Tr15 = 0.0;
    Tr15 = -(q5*Rr1-Tpi5);
    return Tr15;
}

public static double equation7_Tr1(double q, double Rr1, double Tpi){
    double Tr1 = 0.0;
    Tr1 = -(q*Rr1-Tpi);
    return Tr1;
}

public static double equation7_Tpex1(double q1, double Rr2, double Tr11){
    double Tpex1 = 0.0;
    Tpex1 = -(q1*Rr2-Tr11);
    return Tpex1;
}

public static double equation7_Tpex2(double q2, double Rr2, double Tr12){
    double Tpex2 = 0.0;
    Tpex2 = -(q2*Rr2-Tr12);
    return Tpex2;
}

public static double equation7_Tpex3(double q3, double Rr2, double Tr13){
    double Tpex3 = 0.0;
    Tpex3 = -(q3*Rr2-Tr13);
    return Tpex3;
}

public static double equation7_Tpex4(double q4, double Rr2, double Tr14){
    double Tpex4 = 0.0;
    Tpex4 = -(q4*Rr2-Tr14);
    return Tpex4;
}

public static double equation7_Tpex5(double q5, double Rr2, double Tr15){
    double Tpex5 = 0.0;
    Tpex5 = -(q5*Rr2-Tr15);
    return Tpex5;
}

public static double equation7_Tpex(double q, double Rr2, double Tr1){
    double Tpex = 0.0;
    Tpex = -(q*Rr2-Tr1);
}
```



```
        return Tpex;
    }
}

package Fase8;

public class Variables {
    //Variaveis definidas pelo utilizador.
    public static double alpha = 0.25; // limitar ]20%,30%[
    //Resultados das equações
    public static double DeltaP1; // Arrefecimento [Pa]
    public static double DeltaP2; // Calcinação [Pa]
    public static double DeltaP3; // Pré-aquecimento [Pa]
    public static double DeltaP; // Total [Pa]

    public void definirVariaveisUtilizador(double alpha_user){
        alpha = alpha_user;
    }

    public void definirVariaveis(double DeltaP1_user, double DeltaP2_user, double DeltaP3_user,
double DeltaP_user){
        DeltaP1 = DeltaP1_user;
        DeltaP2 = DeltaP2_user;
        DeltaP3 = DeltaP3_user;
        DeltaP = DeltaP_user;
    }

    public double getalpha() {
        return alpha;
    }

    public double getDeltaP1() {
        return DeltaP1;
    }

    public double getDeltaP2() {
        return DeltaP2;
    }

    public double getDeltaP3() {
        return DeltaP3;
    }

    public double getDeltaP() {
        return DeltaP;
    }
}
```

```

}
package Fase8;
import Viewer.Resultados;
public class Equations {
    public static void equation8(){
        Resultados.resultados.append("----- Fase 8: Perda de carga -----");
        ").append(System.getProperty("line.separator"));
        Variables.DeltaP1 = equation8_DeltaP1(Fase3.Variables.Re1, Fase3.Variables.H1,
        Fase1.Variables.dp, Variables.alpha);
        System.out.println("A quantidade de DeltaP1 é: " + Variables.DeltaP1);
        Resultados.resultados.append("A perda de carga na zona de arrefecimento (DeltaP1) é [Pa]:");
        ").append(Variables.DeltaP1).append(System.getProperty("line.separator"));
        Variables.DeltaP2 = equation8_DeltaP2(Fase7.Variables.Re2, Fase2.Variables.H2,
        Fase1.Variables.dp, Variables.alpha);
        System.out.println("A quantidade de DeltaP2 é: " + Variables.DeltaP2);
        Resultados.resultados.append("A perda de carga na zona de calcinação (DeltaP2) é [Pa]:");
        ").append(Variables.DeltaP2).append(System.getProperty("line.separator"));
        Variables.DeltaP3 = equation8_DeltaP3(Fase5.Variables.Re3, Fase5.Variables.H3,
        Fase1.Variables.dp, Variables.alpha);
        System.out.println("A quantidade de DeltaP3 é: " + Variables.DeltaP3);
        Resultados.resultados.append("A perda de carga na zona de pré-aquecimento (DeltaP3) é [Pa]:");
        ").append(Variables.DeltaP3).append(System.getProperty("line.separator"));
        Variables.DeltaP = equation8_DeltaP(Variables.DeltaP1, Variables.DeltaP2, Variables.DeltaP3);
        System.out.println("A perda de carga no forno é (DeltaP): " + Variables.DeltaP);
        Resultados.resultados.append("A perda de carga no forno (DeltaP) é [Pa]:");
        ").append(Variables.DeltaP).append(System.getProperty("line.separator"));
        Resultados.resultados.append("----- Fim da Fase 8 -----");
        ").append(System.getProperty("line.separator")).append(System.getProperty("line.separator"));
    }
    public static double equation8_DeltaP1(double Re1, double H1, double dp, double alpha){
        double DeltaP1 = 0.0;
        DeltaP1 = DeltaP1 =
        ((17.3/Re1+0.336)*(Fase3.Constants.RhoG1*Fase3.Constants.Uf1*H1/(dp*0.001))*(1-
        Fase1.Constants.E)*(Math.pow(Fase1.Constants.E, -
        4.8)))*alpha+((17.3/Re1+0.336)*(Fase3.Constants.RhoG1*Fase3.Constants.Uf1*H1/(dp*0.001))*(1-
        Fase1.Constants.E)*(Math.pow(Fase1.Constants.E, -4.8)));
        return DeltaP1;
    }
}

```

```

public static double equation8_DeltaP2(double Re2, double H2, double dp, double alpha){
    double DeltaP2 = 0.0;

    DeltaP2 = ((17.3/Re2+0.336)*(Fase7.Constants.rhoG2*Fase7.Constants.Uf2*H2/(dp*0.001))*(1-
Fase1.Constants.E)*(Math.pow(Fase1.Constants.E, -
4.8)))*alpha+((17.3/Re2+0.336)*(Fase7.Constants.rhoG2*Fase7.Constants.Uf2*H2/(dp*0.001))*(1-
Fase1.Constants.E)*(Math.pow(Fase1.Constants.E, -4.8)));

    return DeltaP2;
}

public static double equation8_DeltaP3(double Re3, double H3, double dp, double alpha){
    double DeltaP3 = 0.0;

    DeltaP3 = ((17.3/Re3+0.336)*(Fase5.Constants.rhoG3*Fase5.Constants.Uf3*H3/(dp*0.001))*(1-
Fase1.Constants.E)*(Math.pow(Fase1.Constants.E, -
4.8)))*alpha+((17.3/Re3+0.336)*(Fase5.Constants.rhoG3*Fase5.Constants.Uf3*H3/(dp*0.001))*(1-
Fase1.Constants.E)*(Math.pow(Fase1.Constants.E, -4.8)));

    return DeltaP3;
}

public static double equation8_DeltaP(double DeltaP1, double DeltaP2, double DeltaP3){
    double DeltaP = 0.0;

    DeltaP = DeltaP1+DeltaP2+DeltaP3;

    return DeltaP;
}
}

package Fase9;

public class Constants {
    public final static double Pi = 1.0; //[atm]
}

package Fase9;

public class Variables {
    public static double Peq; // Pressão de equilibrio [atm]
    public static double Kd; // fator pré-exponencial [mol/m2 atm s]
    public static double Rc; // Taxa de calcinação [mol/m2]

    public void definirVariaveis (double Peq_user, double Kd_user, double Rc_user){
        Peq = Peq_user;
        Kd = Kd_user;
        Rc = Rc_user;
    }

    public double getPeq() {
        return Peq;
    }
}

```

```

    }
    public double getKd() {
        return Kd;
    }
    public double getRc() {
        return Rc;
    }
}

package Fase9;
import Viewer.Resultados;
public class Equations {
    public static void equation9() {
        Resultados.resultados.append("----- Fase 9: Calcinação -----");
        ").append(System.getProperty("line.separator"));
        Variables.Peq = equation9_Peq(Fase7.Constants.Ti);
        System.out.println("A quantidade de Peq é: " + Variables.Peq);
        Resultados.resultados.append("A pressão de equilíbrio do CO2 (Peq) é [atm]:");
        ").append(Variables.Peq).append(System.getProperty("line.separator"));
        Variables.Kd = equation9_Kd(Fase7.Constants.Ti);
        System.out.println("A quantidade de Kd é: " + Variables.Kd);
        Resultados.resultados.append("Fator pré-exponencial da taxa de decomposição do calcário (Kd)");
        "é [mol/(m2 atm s)]: ").append(Variables.Kd).append(System.getProperty("line.separator"));
        Variables.Rc = equation9_Rc(Variables.Kd, Variables.Peq);
        System.out.println("A quantidade de Rc é: " + Variables.Rc);
        Resultados.resultados.append("A taxa de reação de calcinação (Rc) é [mol/(m2 s)]:");
        ").append(Variables.Rc).append(System.getProperty("line.separator"));
        Resultados.resultados.append("----- Fim da Fase 9 -----");
        ").append(System.getProperty("line.separator")).append(System.getProperty("line.separator"));
    }
    public static double equation9_Peq(double Ti){
        double Peq = 0.0;
        Peq = 4.137E7*(Math.exp(-20474/Ti));
        return Peq;
    }
    public static double equation9_Kd(double Ti){
        double Kd = 0.0;
        Kd = 1.22*(Math.exp(-4026/Ti));
        return Kd;
    }
}

```

```
}  
public static double equation9_Rc(double Kd, double Peq){  
    double Rc = 0.0;  
    Rc = Kd*(Peq-Constants.Pi);  
    return Rc;  
}  
}  
package Fase91;  
public class Variables {  
    public static double hip;  
    public static double Tpar2p; // Temperatura da parede 2ª iteração  
    public static double Tpar3p; // Temperatura da parede 3ª iteração  
    public static double Tf1p; // Temperatura do filme 1ª iteração  
    public static double Tf2p; // Temperatura do filme 2ª iteração  
    public static double Tf3p; // Temperatura do filme 3ª iteração  
    public static double Kamb1p; // Condutibilidade térmica 1ª iteração  
    public static double Kamb2p; // Condutibilidade térmica 2ª iteração  
    public static double Kamb3p; // Condutibilidade térmica 3ª iteração  
    public static double ViscoCin1p; // Viscosidade Cinemática 1ª iteração  
    public static double ViscoCin2p; // Viscosidade Cinemática 2ª iteração  
    public static double ViscoCin3p; // Viscosidade Cinemática 3ª iteração  
    public static double beta1p; // Beta 1ª iteração  
    public static double beta2p; // Beta 2ª iteração  
    public static double beta3p; // Beta 3ª iteração  
    public static double Pr1p; // Prandtl 1ª iteração  
    public static double Pr2p; // Prandtl 2ª iteração  
    public static double Pr3p; // Prandtl 3ª iteração  
    public static double Gr1p; // Grashof 1ª iteração  
    public static double Gr2p; // Grashof 2ª iteração  
    public static double Gr3p; // Grashof 3ª iteração  
    public static double Ra1p; // Rayleigh 1ª iteração  
    public static double Ra2p; // Rayleigh 2ª iteração  
    public static double Ra3p; // Rayleigh 3ª iteração  
    public static double hamb1p; // Coeficiente de convecção do ambiente 1ª iteração  
    public static double hamb2p; // Coeficiente de convecção do ambiente 2ª iteração  
    public static double hamb3p; // Coeficiente de convecção do ambiente 3ª iteração  
    public static double Nu1p; //Nusselt 1ª iteração  
    public static double Nu2p; //Nusselt 2ª iteração
```

```

public static double Nu3p; //Nusselt 3ª iteração
public static double hef1p; // Coeficiente global 1ª iteração
public static double hef2p; // Coeficiente global 2ª iteração
public static double hef3p; // Coeficiente global 3ª iteração
public static double Rip; // Resistência térmica (interior)
public static double Rr1p; // Resistência térmica (Refratário 1)
public static double Rr2p; // Resistência térmica (Refratário 2)
public static double Rex1p; // Resistência térmica (Exterior) 1ª iteração
public static double Rex2p; // Resistência térmica (Exterior) 2ª iteração
public static double Rex3p; // Resistência térmica (Exterior) 3ª iteração
public static double Rt1p; // Resistência térmica total 1ª iteração
public static double Rt2p; // Resistência térmica total 2ª iteração
public static double Rt3p; // Resistência térmica total 3ª iteração
public static double q1p; // Fluxo de calor 1ª iteração
public static double q2p; // Fluxo de calor 2ª iteração
public static double q3p; // Fluxo de calor 3ª iteração
public static double Tpi1p; // Temperatura da parede interior 1ª iteração
public static double Tpi2p; // Temperatura da parede interior 2ª iteração
public static double Tpi3p; // Temperatura da parede interior 3ª iteração
public static double Tr11p; // Temperatura no Refratário 1ª iteração
public static double Tr12p; // Temperatura no Refratário 2ª iteração
public static double Tr13p; // Temperatura no Refratário 3ª iteração
public static double Tpex1p; // Temperatura da parede exterior 1ª iteração
public static double Tpex2p; // Temperatura da parede exterior 2ª iteração
public static double Tpex3p; // Temperatura da parede exterior 3ª iteração

public void definirVariaveis( double hip_user, double Tpar2p_user, double Tpar3p_user, double
Tf1p_user, double Tf2p_user, double Tf3p_user, double Kamb1p_user, double Kamb2p_user, double
Kamb3p_user, double ViscoCin1p_user, double ViscoCin2p_user, double ViscoCin3p_user, double
beta1p_user, double beta2p_user, double beta3p_user, double Pr1p_user, double Pr2p_user, double
Pr3p_user, double Gr1p_user, double Gr2p_user, double Gr3p_user, double Ra1p_user, double
Ra2p_user, double Ra3p_user, double Nu1p_user, double Nu2p_user, double Nu3p_user, double
hamb1p_user, double hamb2p_user, double hamb3p_user, double hef1p_user, double hef2p_user,
double hef3p_user, double Rip_user, double Rr1p_user, double Rr2p_user, double Rex1p_user,
double Rex2p_user, double Rex3p_user, double Rt1p_user, double Rt2p_user, double Rt3p_user,
double q1p_user, double q2p_user, double q3p_user, double Tpi1p_user, double Tpi2p_user, double
Tpi3p_user, double Tr11p_user, double Tr12p_user, double Tr13p_user, double Tpex1p_user, double
Tpex2p_user, double Tpex3p_user){
    hip = hip_user;

```

Tpar2p = Tpar2p_user;
Tpar3p = Tpar3p_user;
Tf1p = Tf1p_user;
Tf2p = Tf2p_user;
Tf3p = Tf3p_user;
Kamb1p = Kamb1p_user;
Kamb2p = Kamb2p_user;
Kamb3p = Kamb3p_user;
ViscoCin1p = ViscoCin1p_user;
ViscoCin2p = ViscoCin2p_user;
ViscoCin3p = ViscoCin3p_user;
beta1p = beta1p_user;
beta2p = beta2p_user;
beta3p = beta3p_user;
Pr1p = Pr1p_user;
Pr2p = Pr2p_user;
Pr3p = Pr3p_user;
Gr1p = Gr1p_user;
Gr2p = Gr2p_user;
Gr3p = Gr3p_user;
Ra1p = Ra1p_user;
Ra2p = Ra2p_user;
Ra3p = Ra3p_user;
Nu1p = Nu1p_user;
Nu2p = Nu2p_user;
Nu3p = Nu3p_user;
hamb1p = hamb1p_user;
hamb2p = hamb2p_user;
hamb3p = hamb3p_user;
hef1p = hef1p_user;
hef2p = hef2p_user;
hef3p = hef3p_user;
Rip = Rip_user;
Rr1p = Rr1p_user;
Rr2p = Rr2p_user;
Rex1p = Rex1p_user;
Rex2p = Rex2p_user;
Rex3p = Rex3p_user;

```
Rt1p = Rt1p_user;
Rt2p = Rt2p_user;
Rt3p = Rt3p_user;
q1p = q1p_user;
q2p = q2p_user;
q3p = q3p_user;
Tpi1p = Tpi1p_user;
Tpi2p = Tpi2p_user;
Tpi3p = Tpi3p_user;
Tr11p = Tr11p_user;
Tr12p = Tr12p_user;
Tr13p = Tr13p_user;
Tpex1p = Tpex1p_user;
Tpex2p = Tpex2p_user;
Tpex3p = Tpex3p_user;
}
public double gethip() {
    return hip;
}
public double getTpar2p() {
    return Tpar2p;
}
public double getTpar3p() {
    return Tpar3p;
}
public double getTf1p() {
    return Tf1p;
}
public double getTf2p() {
    return Tf2p;
}
public double getTf3p() {
    return Tf3p;
}
public double getKamb1p() {
    return Kamb1p;
}
public double getKamb2p() {
```



```
    return Kamb2p;
}
public double getKamb3p() {
    return Kamb3p;
}
public double getViscoCin1p() {
    return ViscoCin1p;
}
public double getViscoCin2p() {
    return ViscoCin2p;
}
public double getViscoCin3p() {
    return ViscoCin3p;
}
public double getbeta1p() {
    return beta1p;
}
public double getbeta2p() {
    return beta2p;
}
public double getbeta3p() {
    return beta3p;
}
public double getPr1p() {
    return Pr1p;
}
public double getPr2p() {
    return Pr2p;
}
public double getPr3p() {
    return Pr3p;
}
public double getGr1p() {
    return Gr1p;
}
public double getGr2p() {
    return Gr2p;
}
```

```
public double getGr3p() {  
    return Gr3p;  
}  
public double getRa1p() {  
    return Ra1p;  
}  
public double getRa2p() {  
    return Ra2p;  
}  
public double getRa3p() {  
    return Ra3p;  
}  
public double getNu1p() {  
    return Nu1p;  
}  
public double getNu2p() {  
    return Nu2p;  
}  
public double getNu3p() {  
    return Nu3p;  
}  
public double gethamb1p() {  
    return hamb1p;  
}  
public double gethamb2p() {  
    return hamb2p;  
}  
public double gethamb3p() {  
    return hamb3p;  
}  
public double gethef1p() {  
    return hef1p;  
}  
public double gethef2p() {  
    return hef2p;  
}  
public double gethef3p() {  
    return hef3p;  
}
```

```
}  
public double getRip() {  
    return Rip;  
}  
public double getRr1p() {  
    return Rr1p;  
}  
public double getRr2p() {  
    return Rr2p;  
}  
public double getRex1p() {  
    return Rex1p;  
}  
public double getRex2p() {  
    return Rex2p;  
}  
public double getRex3p() {  
    return Rex3p;  
}  
public double getRt1p() {  
    return Rt1p;  
}  
public double getRt2p() {  
    return Rt2p;  
}  
public double getRt3p() {  
    return Rt3p;  
}  
public double getq1p() {  
    return q1p;  
}  
public double getq2p() {  
    return q2p;  
}  
public double getq3p() {  
    return q3p;  
}  
public double getTpi1p() {
```

```
    return Tpi1p;
}
public double getTpi2p() {
    return Tpi2p;
}
public double getTpi3p() {
    return Tpi3p;
}
public double getTr11p() {
    return Tr11p;
}
public double getTr12p() {
    return Tr12p;
}
public double getTr13p() {
    return Tr13p;
}
public double getTpex1p() {
    return Tpex1p;
}
public double getTpex2p() {
    return Tpex2p;
}
public double getTpex3p() {
    return Tpex3p;
}
}
package Fase91;
import Viewer.Resultados;
public class Equations {
    public static void equation91(){
        Resultados.resultados.append("----- Fase 9.1: Análise transferência de calor (zona pré-
aquecimento) -----").append(System.getProperty("line.separator"));
        Variables.hip = equation91_hip(Fase5.Variables.Re3);
        System.out.println("A quantidade de hip é: " + Variables.hip);
        Resultados.resultados.append("O coeficiente de transferência de calor no interior do forno (hip) é
[W/m2K]: ").append(Variables.hip).append(System.getProperty("line.separator"));
        Variables.Tf1p = equation91_Tf1p(Fase7.Variables.Tamb);
```

```

System.out.println("A quantidade de Tf1p é: " + Variables.Tf1p);
Resultados.resultados.append("A temperatura do filme da 1ª iteração (Tf1p) é [°C]:
").append(Variables.Tf1p).append(System.getProperty("line.separator"));
Variables.Kamb1p = equation91_Kamb1p(Variables.Tf1p);
System.out.println("A quantidade de Kamb1p é: " + Variables.Kamb1p);
Resultados.resultados.append("Condutividade térmica do ar ambiente 1ª iteração (Kamb1p) é
[W/mK]: ").append(Variables.Kamb1p).append(System.getProperty("line.separator"));
Variables.ViscoCin1p = equation91_ViscoCin1p(Variables.Tf1p);
System.out.println("A quantidade de ViscoCin1p é: " + Variables.ViscoCin1p);
Resultados.resultados.append("A viscosidade cinemática do ar da 1ª iteração (ViscoCin1p) é
[m2/s]: ").append(Variables.ViscoCin1p).append(System.getProperty("line.separator"));
Variables.beta1p = equation91_beta1p(Variables.Tf1p);
System.out.println("A quantidade de beta1p é: " + Variables.beta1p);
Resultados.resultados.append("Coeficiente de expansão volumétrica da 1ª iteração (beta1p) é
[1/K]: ").append(Variables.beta1p).append(System.getProperty("line.separator"));
Variables.Pr1p = equation91_Pr1p(Variables.Tf1p);
System.out.println("A quantidade de Pr1p é: " + Variables.Pr1p);
Resultados.resultados.append("O número de Prandtl da 1ª iteração (Pr1p) é:
").append(Variables.Pr1p).append(System.getProperty("line.separator"));
Variables.Gr1p = equation91_Gr1p(Variables.beta1p, Fase7.Variables.Tamb,
Fase7.Variables.Lc, Variables.ViscoCin1p);
System.out.println("A quantidade de Gr1p é: " + Variables.Gr1p);
Resultados.resultados.append("O número de Grashof da 1ª iteração (Gr1p) é:
").append(Variables.Gr1p).append(System.getProperty("line.separator"));
Variables.Ra1p = equation91_Ra1p(Variables.Gr1p, Variables.Pr1p);
System.out.println("A quantidade de Ra1p é: " + Variables.Ra1p);
Resultados.resultados.append("O número de Rayleigh da 1ª iteração (Ra1p) é:
").append(Variables.Ra1p).append(System.getProperty("line.separator"));
Variables.Nu1p = equation91_Nu1p(Variables.Ra1p, Variables.Pr1p);
System.out.println("A quantidade de Nu1p é: " + Variables.Nu1p);
Resultados.resultados.append("O número de Nusselt da 1ª iteração (Nu1p) é:
").append(Variables.Nu1p).append(System.getProperty("line.separator"));
Variables.hamb1p = equation91_hamb1p(Variables.Nu1p, Variables.Kamb1p,
Fase7.Variables.Lc);
System.out.println("A quantidade de hamb1p é: " + Variables.hamb1p);
Resultados.resultados.append("O coeficiente de convecção do ambiente da 1ª iteração (hamb1p)
é [W/m2K]: ").append(Variables.hamb1p).append(System.getProperty("line.separator"));
Variables.hef1p = equation91_hef1p(Variables.hamb1p, Fase7.Variables.Tamb);

```

```

System.out.println("A quantidade de hef1p é: " + Variables.hef1p);
Resultados.resultados.append("O coeficiente de global (efetivo) da 1ª iteração (hef1p) é
[W/m2K]: ").append(Variables.hef1p).append(System.getProperty("line.separator"));
Variables.Rip = equation91_Rip(Variables.hip, Fase7.Variables.Lc);
System.out.println("A quantidade de Rip é: " + Variables.Rip);
Variables.Rr1p = equation91_Rr1p(Fase7.Variables.r1, Fase1.Variables.R, Fase7.Variables.Lc);
System.out.println("A quantidade de Rr1p é: " + Variables.Rr1p);
Variables.Rr2p = equation91_Rr2p(Fase7.Variables.r1, Fase7.Variables.r2, Fase7.Variables.Lc);
System.out.println("A quantidade de Rr2p é: " + Variables.Rr2p);
Variables.Rex1p = equation91_Rex1p(Variables.hef1p, Fase7.Variables.Lc);
System.out.println("A quantidade de Rex1p é: " + Variables.Rex1p);
Variables.Rt1p = equation91_Rt1p(Variables.Rip, Variables.Rr1p, Variables.Rr2p,
Variables.Rex1p);
System.out.println("A quantidade de Rt1p é: " + Variables.Rt1p);
Resultados.resultados.append("Resistência térmica total da 1ª iteração (Rt1p) é [m2/W K]:
").append(Variables.Rt1p).append(System.getProperty("line.separator"));
Variables.q1p = equation91_q1p(Fase7.Variables.Tamb, Variables.Rt1p);
System.out.println("A quantidade de q1p é: " + Variables.q1p);
Resultados.resultados.append("O fluxo de calor da 1ª iteração (q1p) é [W/m2]:
").append(Variables.q1p).append(System.getProperty("line.separator"));
Variables.Tpi1p = equation91_Tpi1p(Variables.q1p, Variables.Rip);
System.out.println("A quantidade de Tpi1p é: " + Variables.Tpi1p);
Variables.Tr11p = equation91_Tr11p(Variables.q1p, Variables.Rr1p, Variables.Tpi1p);
System.out.println("A quantidade de Tr11p é: " + Variables.Tr11p);
Variables.Tpex1p = equation91_Tpex1p(Variables.q1p, Variables.Rr2p, Variables.Tr11p);
System.out.println("A quantidade de Tpex1 é: " + Variables.Tpex1p);
Resultados.resultados.append("A temperatura da parede exterior na 1ª iteração (Tpex1p) é [°C]:
").append(Variables.Tpex1p).append(System.getProperty("line.separator"));
Variables.Tpar2p = equation91_Tpar2p(Variables.Tpex1p);
System.out.println("A quantidade de Tpar2p é: " + Variables.Tpar2p);
Variables.Tf2p = equation91_Tf2p(Variables.Tpar2p, Fase7.Variables.Tamb);
System.out.println("A quantidade de Tf2p é: " + Variables.Tf2p);
Resultados.resultados.append("A temperatura do filme da 2ª iteração (Tf2p) é [°C]:
").append(Variables.Tf2p).append(System.getProperty("line.separator"));
Variables.Kamb2p = equation91_Kamb2p(Variables.Tf2p);
System.out.println("A quantidade de Kamb2p é: " + Variables.Kamb2p);
Resultados.resultados.append("Condutividade térmica do ar ambiente 2ª iteração (Kamb2p) é
[W/mK]: ").append(Variables.Kamb2p).append(System.getProperty("line.separator"));

```

```

Variables.ViscoCin2p = equation91_ViscoCin2p(Variables.Tf2p);
System.out.println("A quantidade de ViscoCin2p é: " + Variables.ViscoCin2p);
Resultados.resultados.append("A viscosidade cinemática do ar da 2ª iteração (ViscoCin2p) é
[m2/s]: ").append(Variables.ViscoCin2p).append(System.getProperty("line.separator"));
Variables.beta2p = equation91_beta2p(Variables.Tf2p);
System.out.println("A quantidade de beta2p é: " + Variables.beta2p);
Resultados.resultados.append("Coeficiente de expansão volumétrica da 2ª iteração (beta2p) é
[1/K]: ").append(Variables.beta2p).append(System.getProperty("line.separator"));
Variables.Pr2p = equation91_Pr2p(Variables.Tf2p);
System.out.println("A quantidade de Pr2p é: " + Variables.Pr2p);
Resultados.resultados.append("O número de Prandtl da 2ª iteração (Pr2p) é:
").append(Variables.Pr2p).append(System.getProperty("line.separator"));
Variables.Gr2p = equation91_Gr2p(Variables.beta2p, Fase7.Variables.Tpar2,
Fase7.Variables.Tamb, Fase7.Variables.Lc, Variables.ViscoCin2p);
System.out.println("A quantidade de Gr2p é: " + Variables.Gr2p);
Resultados.resultados.append("O número de Grashof da 2ª iteração (Gr2p) é:
").append(Variables.Gr2p).append(System.getProperty("line.separator"));
Variables.Ra2p = equation91_Ra2p(Variables.Gr2p, Variables.Pr2p);
System.out.println("A quantidade de Ra2p é: " + Variables.Ra2p);
Resultados.resultados.append("O número de Rayleigh da 2ª iteração (Ra2p) é:
").append(Variables.Ra2p).append(System.getProperty("line.separator"));
Variables.Nu2p = equation91_Nu2p(Variables.Ra2p, Variables.Pr2p);
System.out.println("A quantidade de Nu2p é: " + Variables.Nu2p);
Resultados.resultados.append("O número de Nusselt da 2ª iteração (Nu2p) é:
").append(Variables.Nu2p).append(System.getProperty("line.separator"));
Variables.hamb2p = equation91_hamb2p(Variables.Nu2p, Variables.Kamb2p,
Fase7.Variables.Lc);
System.out.println("A quantidade de hamb2p é: " + Variables.hamb2p);
Resultados.resultados.append("O coeficiente de convecção do ambiente da 2ª iteração (hamb2p)
é [W/m2K]: ").append(Variables.hamb2p).append(System.getProperty("line.separator"));
Variables.hef2p = equation91_hef2p(Variables.hamb2p, Variables.Tpar2,
Fase7.Variables.Tamb);
System.out.println("A quantidade de hef2p é: " + Variables.hef2p);
Resultados.resultados.append("O coeficiente de global (efetivo) da 2ª iteração (hef2p) é
[W/m2K]: ").append(Variables.hef2p).append(System.getProperty("line.separator"));
Variables.Rex2p = equation91_Rex2p(Variables.hef2p, Fase7.Variables.Lc);
System.out.println("A quantidade de Rex2p é: " + Variables.Rex2p);

```

```

Variables.Rt2p = equation91_Rt2p(Variables.Rip, Variables.Rr1p, Variables.Rr2p,
Variables.Rex2p);
System.out.println("A quantidade de Rt2 é: " + Variables.Rt2p);
Resultados.resultados.append("Resistência térmica total da 2ª iteração (Rt2p) é [m2/W K]:
").append(Variables.Rt2p).append(System.getProperty("line.separator"));
Variables.q2p = equation91_q2p(Fase7.Variables.Tamb, Variables.Rt2p);
System.out.println("A quantidade de q2p é: " + Variables.q2p);
Resultados.resultados.append("O fluxo de calor da 2ª iteração (q2p) é [W/m2]:
").append(Variables.q2p).append(System.getProperty("line.separator"));
Variables.Tpi2p = equation91_Tpi2p(Variables.q2p, Variables.Rip);
System.out.println("A quantidade de Tpi2p é: " + Variables.Tpi2p);
Variables.Tr12p = equation91_Tr12p(Variables.q2p, Variables.Rr1p, Variables.Tpi2p);
System.out.println("A quantidade de Tr12p é: " + Variables.Tr12p);
Variables.Tpex2p = equation91_Tpex2p(Variables.q2p, Variables.Rr2p, Variables.Tr12p);
System.out.println("A quantidade de Tpex2 é: " + Variables.Tpex2p);
Resultados.resultados.append("A temperatura da parede exterior na 2ª iteração (Tpex2p) é [°C]:
").append(Variables.Tpex2p).append(System.getProperty("line.separator"));
Variables.Tpar3p = equation91_Tpar3p(Variables.Tpex2p);
System.out.println("A quantidade de Tpar3p é: " + Variables.Tpar3p);
Variables.Tf3p = equation91_Tf3p(Variables.Tpar3p, Fase7.Variables.Tamb);
System.out.println("A quantidade de Tf3p é: " + Variables.Tf3p);
Resultados.resultados.append("A temperatura do filme da 3ª iteração (Tf3p) é [°C]:
").append(Variables.Tf3p).append(System.getProperty("line.separator"));
Variables.Kamb3p = equation91_Kamb3p(Variables.Tf3p);
System.out.println("A quantidade de Kamb3p é: " + Variables.Kamb3p);
Resultados.resultados.append("Condutividade térmica do ar ambiente 3ª iteração (Kamb3p) é
[W/mK]: ").append(Variables.Kamb3p).append(System.getProperty("line.separator"));
Variables.ViscoCin3p = equation91_ViscoCin3p(Variables.Tf3p);
System.out.println("A quantidade de ViscoCin3p é: " + Variables.ViscoCin3p);
Resultados.resultados.append("A viscosidade cinemática do ar da 3ª iteração (ViscoCin3p) é
[m2/s]: ").append(Variables.ViscoCin3p).append(System.getProperty("line.separator"));
Variables.beta3p = equation91_beta3p(Variables.Tf3p);
System.out.println("A quantidade de beta3p é: " + Variables.beta3p);
Resultados.resultados.append("Coeficiente de expansão volumétrica da 3ª iteração (beta3p) é
[1/K]: ").append(Variables.beta3p).append(System.getProperty("line.separator"));
Variables.Pr3p = equation91_Pr3p(Variables.Tf3p);
System.out.println("A quantidade de Pr3p é: " + Variables.Pr3p);

```



```

Resultados.resultados.append("O número de Prandtl da 3ª iteração (Pr3p) é:
").append(Variables.Pr3p).append(System.getProperty("line.separator"));
Variables.Gr3p = equation91_Gr3p(Variables.beta3p, Fase7.Variables.Tpar3,
Fase7.Variables.Tamb, Fase7.Variables.Lc, Variables.ViscoCin3p);
System.out.println("A quantidade de Gr3p é: " + Variables.Gr3p);
Resultados.resultados.append("O número de Grashof da 3ª iteração (Gr3p) é:
").append(Variables.Gr3p).append(System.getProperty("line.separator"));
Variables.Ra3p = equation91_Ra3p(Variables.Gr3p, Variables.Pr3p);
System.out.println("A quantidade de Ra3p é: " + Variables.Ra3p);
Resultados.resultados.append("O número de Rayleigh da 3ª iteração (Ra3p) é:
").append(Variables.Ra3p).append(System.getProperty("line.separator"));
Variables.Nu3p = equation91_Nu3p(Variables.Ra3p, Variables.Pr3p);
System.out.println("A quantidade de Nu3p é: " + Variables.Nu3p);
Resultados.resultados.append("O número de Nusselt da 3ª iteração (Nu3p) é:
").append(Variables.Nu3p).append(System.getProperty("line.separator"));
Variables.hamb3p = equation91_hamb3p(Variables.Nu3p, Variables.Kamb3p,
Fase7.Variables.Lc);
System.out.println("A quantidade de hamb3p é: " + Variables.hamb3p);
Resultados.resultados.append("O coeficiente de convecção do ambiente da 3ª iteração (hamb3p)
é [W/m2K]: ").append(Variables.hamb3p).append(System.getProperty("line.separator"));
Variables.hef3p = equation91_hef3p(Variables.hamb3p, Variables.Tpar3p,
Fase7.Variables.Tamb);
System.out.println("A quantidade de hef3p é: " + Variables.hef3p);
Resultados.resultados.append("O coeficiente de global (efetivo) da 3ª iteração (hef3p) é
[W/m2K]: ").append(Variables.hef3p).append(System.getProperty("line.separator"));
Variables.Rex3p = equation91_Rex3p(Variables.hef3p, Fase7.Variables.Lc);
System.out.println("A quantidade de Rex3p é: " + Variables.Rex3p);
Variables.Rt3p = equation91_Rt3p(Variables.Rip, Variables.Rr1p, Variables.Rr2p,
Variables.Rex3p);
System.out.println("A quantidade de Rt3p é: " + Variables.Rt3p);
Resultados.resultados.append("Resistência térmica total da 3ª iteração (Rt3p) é [m2/W K]:
").append(Variables.Rt3p).append(System.getProperty("line.separator"));
Variables.q3p = equation91_q3p(Fase7.Variables.Tamb, Variables.Rt3p);
System.out.println("A quantidade de q3p é: " + Variables.q3p);
Resultados.resultados.append("O fluxo de calor da 3ª iteração (q3p) é [W/m2]:
").append(Variables.q3p).append(System.getProperty("line.separator"));
Variables.Tpi3p = equation91_Tpi3p(Variables.q3p, Variables.Rip);
System.out.println("A quantidade de Tpi3p é: " + Variables.Tpi3p);

```

```
Variables.Tr13p = equation91_Tr13p(Variables.q3p, Variables.Rr1p, Variables.Tpi3p);
System.out.println("A quantidade de Tr13p é: " + Variables.Tr13p);
Variables.Tpex3p = equation91_Tpex3p(Variables.q3p, Variables.Rr2p, Variables.Tr13p);
System.out.println("A temperatura da parede na zona de pré-aquecimento é (Tpex3p): " +
Variables.Tpex3p);
Resultados.resultados.append("A temperatura da parede exterior na zona de pré-aquecimento
(Tpex3p) é [°C]: ").append(Variables.Tpex3p).append(System.getProperty("line.separator"));

Resultados.resultados.append("----- Fim da Fase 9.1 -----
").append(System.getProperty("line.separator")).append(System.getProperty("line.separator"));
}
public static double equation91_hip(double Re3){
    double hip = 0.0;
    hip = 0.525*(Math.pow(Re3,0.75));
    return hip;
}
public static double equation91_Tpar2p(double Tpex1p){
    double Tpar2p = 0.0;
    Tpar2p = Tpex1p;
    return Tpar2p;
}
public static double equation91_Tpar3p(double Tpex2p){
    double Tpar3p = 0.0;
    Tpar3p = Tpex2p;
    return Tpar3p;
}

public static double equation91_Tf1p(double Tamb){
    double Tf1p = 0.0;
    Tf1p = (Fase7.Constants.Tpar1+Tamb)/2;
    return Tf1p;
}
public static double equation91_Tf2p(double Tpar2p, double Tamb){
    double Tf2 = 0.0;
    Tf2 = (Tpar2p+Tamb)/2;
    return Tf2;
}
public static double equation91_Tf3p(double Tpar3, double Tamb){
```

```
double Tf3p = 0.0;
Tf3p = (Tpar3+Tamb)/2;
return Tf3p;
}

public static double equation91_Kamb1p(double Tf1p){
double Kamb1p = 0.0;
Kamb1p = -1.1198897E-20*Math.pow(Tf1p,6)+8.7864215E-15*Math.pow(Tf1p,5)-4.4374479E-
14*Math.pow(Tf1p,4)+9.6559189E-10*Math.pow(Tf1p,3)-6.7839993E-
8*Math.pow(Tf1p,2)+7.1477994E-5*Tf1p+2.4295037E-2;
return Kamb1p;
}

public static double equation91_Kamb2p(double Tf2p){
double Kamb2p = 0.0;
Kamb2p = -1.1198897E-20*Math.pow(Tf2p,6)+8.7864215E-15*Math.pow(Tf2p,5)-4.4374479E-
14*Math.pow(Tf2p,4)+9.6559189E-10*Math.pow(Tf2p,3)-6.7839993E-
8*Math.pow(Tf2p,2)+7.1477994E-5*Tf2p+2.4295037E-2;
return Kamb2p;
}

public static double equation91_Kamb3p(double Tf3p){
double Kamb3p = 0.0;
Kamb3p = -1.1198897E-20*Math.pow(Tf3p,6)+8.7864215E-15*Math.pow(Tf3p,5)-4.4374479E-
14*Math.pow(Tf3p,4)+9.6559189E-10*Math.pow(Tf3p,3)-6.7839993E-
8*Math.pow(Tf3p,2)+7.1477994E-5*Tf3p+2.4295037E-2;
return Kamb3p;
}

public static double equation91_ViscoCin1p(double Tf1p){
double ViscoCin1p = 0.0;
ViscoCin1p = 4.365140866895E-20*Math.pow(Tf1p,6)-4.039290616056E-
17*Math.pow(Tf1p,5)+1.088015085567E-14*Math.pow(Tf1p,4)-1.141636784366E-
12*Math.pow(Tf1p,3)+1.444436360192E-10*Math.pow(Tf1p,2)+8.699748434118E-
8*Tf1p+1.331422047548E-5;
return ViscoCin1p;
}

public static double equation91_ViscoCin2p(double Tf2p){
double ViscoCin2p = 0.0;
ViscoCin2p = 4.365140866895E-20*Math.pow(Tf2p,6)-4.039290616056E-
17*Math.pow(Tf2p,5)+1.088015085567E-14*Math.pow(Tf2p,4)-1.141636784366E-
```

```

12*Math.pow(Tf2p,3)+1.444436360192E-10*Math.pow(Tf2p,2)+8.699748434118E-
8*Tf2p+1.331422047548E-5;
    return ViscoCin2p;
}
public static double equation91_ViscoCin3p(double Tf3p){
    double ViscoCin3p = 0.0;
    ViscoCin3p = 4.365140866895E-20*Math.pow(Tf3p,6)-4.039290616056E-
17*Math.pow(Tf3p,5)+1.088015085567E-14*Math.pow(Tf3p,4)-1.141636784366E-
12*Math.pow(Tf3p,3)+1.444436360192E-10*Math.pow(Tf3p,2)+8.699748434118E-
8*Tf3p+1.331422047548E-5;
    return ViscoCin3p;
}
public static double equation91_beta1p(double Tf1p){
    double beta1p = 0.0;
    beta1p = 5.8070940E-17*Math.pow(Tf1p,6)-2.9994199E-14*Math.pow(Tf1p,5)+5.1941928E-
12*Math.pow(Tf1p,4)-3.7374860E-10*Math.pow(Tf1p,3)+3.9708796E-8*Math.pow(Tf1p,2)-
1.2953005E-5*Tf1p+3.6734906E-3;
    return beta1p;
}
public static double equation91_beta2p(double Tf2p){
    double beta2p = 0.0;
    beta2p = 5.8070940E-17*Math.pow(Tf2p,6)-2.9994199E-14*Math.pow(Tf2p,5)+5.1941928E-
12*Math.pow(Tf2p,4)-3.7374860E-10*Math.pow(Tf2p,3)+3.9708796E-8*Math.pow(Tf2p,2)-
1.2953005E-5*Tf2p+3.6734906E-3;
    return beta2p;
}
public static double equation91_beta3p(double Tf3p){
    double beta3p = 0.0;
    beta3p = 5.8070940E-17*Math.pow(Tf3p,6)-2.9994199E-14*Math.pow(Tf3p,5)+5.1941928E-
12*Math.pow(Tf3p,4)-3.7374860E-10*Math.pow(Tf3p,3)+3.9708796E-8*Math.pow(Tf3p,2)-
1.2953005E-5*Tf3p+3.6734906E-3;
    return beta3p;
}
public static double equation91_Pr1p(double Tf1p){
    double Pr1p = 0.0;
    Pr1p = -8.9663606E-15*Math.pow(Tf1p,6)+4.2115168E-12*Math.pow(Tf1p,5)-5.8250457E-
10*Math.pow(Tf1p,4)+8.6537272E-9*Math.pow(Tf1p,3)+2.4053808E-6*Math.pow(Tf1p,2)-
1.9350745E-4*Tf1p+7.1547018E-1;

```

```

    return Pr1p;
}

public static double equation91_Pr2p(double Tf2p){
    double Pr2p = 0.0;

    Pr2p = -8.9663606E-15*Math.pow(Tf2p,6)+4.2115168E-12*Math.pow(Tf2p,5)-5.8250457E-
10*Math.pow(Tf2p,4)+8.6537272E-9*Math.pow(Tf2p,3)+2.4053808E-6*Math.pow(Tf2p,2)-
1.9350745E-4*Tf2p+7.1547018E-1;

    return Pr2p;
}

public static double equation91_Pr3p(double Tf3p){
    double Pr3p = 0.0;

    Pr3p = -8.9663606E-15*Math.pow(Tf3p,6)+4.2115168E-12*Math.pow(Tf3p,5)-5.8250457E-
10*Math.pow(Tf3p,4)+8.6537272E-9*Math.pow(Tf3p,3)+2.4053808E-6*Math.pow(Tf3p,2)-
1.9350745E-4*Tf3p+7.1547018E-1;

    return Pr3p;
}

public static double equation91_Gr1p(double beta1p, double Tamb, double Lc, double
ViscoCin1p){
    double Gr1p = 0.0;

    Gr1p = (Fase7.Constants.g*beta1p*(Fase7.Constants.Tpar1-Tamb)*(Math.pow(Lc,
3)))/(Math.pow(ViscoCin1p, 2));

    return Gr1p;
}

public static double equation91_Gr2p(double beta2p, double Tpar2p, double Tamb, double Lc,
double ViscoCin2p){
    double Gr2p = 0.0;

    Gr2p = (Fase7.Constants.g*beta2p*(Tpar2p-Tamb)*(Math.pow(Lc, 3)))/(Math.pow(ViscoCin2p,
2));

    return Gr2p;
}

public static double equation91_Gr3p(double beta3p, double Tpar3p, double Tamb, double Lc,
double ViscoCin3p){
    double Gr3p = 0.0;

    Gr3p = (Fase7.Constants.g*beta3p*(Tpar3p-Tamb)*(Math.pow(Lc, 3)))/(Math.pow(ViscoCin3p,
2));

    return Gr3p;
}

public static double equation91_Ra1p(double Gr1p, double Pr1p){

```

```
double Ra1p = 0.0;
Ra1p = Gr1p*Pr1p;
return Ra1p;
}

public static double equation91_Ra2p(double Gr2p, double Pr2p){
double Ra2p = 0.0;
Ra2p = Gr2p*Pr2p;
return Ra2p;
}

public static double equation91_Ra3p(double Gr3p, double Pr3p){
double Ra3p = 0.0;
Ra3p = Gr3p*Pr3p;
return Ra3p;
}

public static double equation91_Nu1p(double Ra1p, double Pr1p){
double Nu1p = 0.0;
Nu1p =
Math.pow(0.825+0.387*(Math.pow(Ra1p,0.166666667))/((Math.pow(1+(Math.pow(0.492/Pr1p,0.5625))
,0.296296296))),2);
return Nu1p;
}

public static double equation91_Nu2p(double Ra2p, double Pr2p){
double Nu2p = 0.0;
Nu2p =
Math.pow(0.825+0.387*(Math.pow(Ra2p,0.166666667))/((Math.pow(1+(Math.pow(0.492/Pr2p,0.5625))
,0.296296296))),2);
return Nu2p;
}

public static double equation91_Nu3p(double Ra3p, double Pr3p){
double Nu3p = 0.0;
Nu3p =
Math.pow(0.825+0.387*(Math.pow(Ra3p,0.166666667))/((Math.pow(1+(Math.pow(0.492/Pr3p,0.5625))
,0.296296296))),2);
return Nu3p;
}

public static double equation91_hamb1p(double Nu1p, double Kamb1p, double Lc){
double hamb1p = 0.0;
hamb1p = Nu1p*Kamb1p/Lc;
```

```
    return hamb1p;
}

public static double equation91_hamb2p(double Nu2p, double Kamb2p, double Lc){
    double hamb2p = 0.0;
    hamb2p = Nu2p*Kamb2p/Lc;
    return hamb2p;
}

public static double equation91_hamb3p(double Nu3p, double Kamb3p, double Lc){
    double hamb3p = 0.0;
    hamb3p = Nu3p*Kamb3p/Lc;
    return hamb3p;
}

public static double equation91_hef1p(double hamb1p, double Tamb){
    double hef1p = 0.0;
    hef1p = (hamb1p*(Fase7.Constants.Tpar1-
Tamb)+1*Fase7.Constants.Emi*Fase7.Constants.sigma*((Math.pow(Fase7.Constants.Tpar1+273.15,4
))-(Math.pow(Tamb+273.15,4))))/(Fase7.Constants.Tpar1-Tamb);
    return hef1p;
}

public static double equation91_hef2p(double hamb2p, double Tpar2p, double Tamb){
    double hef2p = 0.0;
    hef2p = (hamb2p*(Tpar2p-
Tamb)+1*Fase7.Constants.Emi*Fase7.Constants.sigma*((Math.pow(Tpar2p+273.15,4))-
(Math.pow(Tamb+273.15,4))))/(Tpar2p-Tamb);
    return hef2p;
}

public static double equation91_hef3p(double hamb3p, double Tpar3p, double Tamb){
    double hef3p = 0.0;
    hef3p = (hamb3p*(Tpar3p-
Tamb)+1*Fase7.Constants.Emi*Fase7.Constants.sigma*((Math.pow(Tpar3p+273.15,4))-
(Math.pow(Tamb+273.15,4))))/(Tpar3p-Tamb);
    return hef3p;
}

public static double equation91_Rip(double hip, double Lc){
    double Rip = 0.0;
    Rip = 1/(2*Math.PI*hip*Lc);
    return Rip;
}
```

```
public static double equation91_Rr1p(double r1, double R, double Lc){
    double Rr1p = 0.0;
    Rr1p = (Math.log(r1/Fase1.Variables.R))/(2*Math.PI*Fase7.Constants.Kr1*Lc);
    return Rr1p;
}

public static double equation91_Rr2p(double r1, double r2, double Lc){
    double Rr2p = 0.0;
    Rr2p = (Math.log(r2/r1))/(2*Math.PI*Fase7.Constants.Kr2*Lc);
    return Rr2p;
}

public static double equation91_Rex1p(double hef1p, double Lc){
    double Rex1p = 0.0;
    Rex1p = 1/(2*Math.PI*hef1p*Lc);
    return Rex1p;
}

public static double equation91_Rex2p(double hef2p, double Lc){
    double Rex2p = 0.0;
    Rex2p = 1/(2*Math.PI*hef2p*Lc);
    return Rex2p;
}

public static double equation91_Rex3p(double hef3p, double Lc){
    double Rex3p = 0.0;
    Rex3p = 1/(2*Math.PI*hef3p*Lc);
    return Rex3p;
}

public static double equation91_Rt1p(double Rip, double Rr1p, double Rr2p, double Rex1p){
    double Rt1p = 0.0;
    Rt1p = Rip+Rr1p+Rr2p+Rex1p;
    return Rt1p;
}

public static double equation91_Rt2p(double Rip, double Rr1p, double Rr2p, double Rex2p){
    double Rt2p = 0.0;
    Rt2p = Rip+Rr1p+Rr2p+Rex2p;
    return Rt2p;
}

public static double equation91_Rt3p(double Rip, double Rr1p, double Rr2p, double Rex3p){
    double Rt3p = 0.0;
    Rt3p = Rip+Rr1p+Rr2p+Rex3p;
```



```
    return Rt3p;
}

public static double equation91_q1p(double Tamb, double Rt1p){
    double q1p = 0.0;
    q1p = (Fase5.Constants.Tip-Tamb)/Rt1p;
    return q1p;
}

public static double equation91_q2p(double Tamb, double Rt2p){
    double q2p = 0.0;
    q2p = (Fase5.Constants.Tip-Tamb)/Rt2p;
    return q2p;
}

public static double equation91_q3p(double Tamb, double Rt3p){
    double q3p = 0.0;
    q3p = (Fase5.Constants.Tip-Tamb)/Rt3p;
    return q3p;
}

public static double equation91_Tpi1p(double q1p, double Rip){
    double Tpi1p = 0.0;
    Tpi1p = - (q1p*Rip-Fase5.Constants.Tip);
    return Tpi1p;
}

public static double equation91_Tpi2p(double q2p, double Rip){
    double Tpi2p = 0.0;
    Tpi2p = - (q2p*Rip-Fase5.Constants.Tip);
    return Tpi2p;
}

public static double equation91_Tpi3p(double q3p, double Rip){
    double Tpi3p = 0.0;
    Tpi3p = - (q3p*Rip-Fase5.Constants.Tip);
    return Tpi3p;
}

public static double equation91_Tr11p(double q1p, double Rr1p, double Tpi1p){
    double Tr11p = 0.0;
    Tr11p = -(q1p*Rr1p-Tpi1p);
    return Tr11p;
}
```

```
public static double equation91_Tr12p(double q2p, double Rr1p, double Tpi2p){
    double Tr12p = 0.0;
    Tr12p = -(q2p*Rr1p-Tpi2p);
    return Tr12p;
}

public static double equation91_Tr13p(double q3p, double Rr1p, double Tpi3p){
    double Tr13p = 0.0;
    Tr13p = -(q3p*Rr1p-Tpi3p);
    return Tr13p;
}

public static double equation91_Tpex1p(double q1p, double Rr2p, double Tr11p){
    double Tpex1p = 0.0;
    Tpex1p = -(q1p*Rr2p-Tr11p);
    return Tpex1p;
}

public static double equation91_Tpex2p(double q2p, double Rr2p, double Tr12p){
    double Tpex2p = 0.0;
    Tpex2p = -(q2p*Rr2p-Tr12p);
    return Tpex2p;
}

public static double equation91_Tpex3p(double q3p, double Rr2p, double Tr13p){
    double Tpex3p = 0.0;
    Tpex3p = -(q3p*Rr2p-Tr13p);
    return Tpex3p;
}
}

package Viewer;
import Fase1.Equations;
import Fase1.Variables;
import java.awt.Dimension;
import java.awt.Toolkit;
import javax.swing.JOptionPane;
import javax.swing.event.DocumentEvent;
import javax.swing.event.DocumentListener;
public class Principal extends javax.swing.JFrame {
    private String error = "";
    private int quantidadeDeCaO_auxInt;
    private double quantidadeDeCaCO3_auxDouble;
```

```
private double R_auxDouble;
private double dp_auxDouble;
private int Tamb_auxInt;
private int Tge1_auxInt;
private int Tps1_auxInt;
private double e_auxDouble;
public Principal() {
    this.setTitle("FornoCal");
    this.setResizable(false);
    initComponents();
    setListener();
    Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
    Dimension frameSize = this.getSize();
    if (frameSize.height > screenSize.height) {
        frameSize.height = screenSize.height;
    }
    if (frameSize.width > screenSize.width) {
        frameSize.width = screenSize.width;
    }
    this.setLocation( (screenSize.width - frameSize.width)/2, (screenSize.height -
frameSize.height)/2);
}
private void setListener(){
    Tamb.getDocument().addDocumentListener(new DocumentListener() {
        public void changedUpdate(DocumentEvent e) {
            act();
        }
        public void removeUpdate(DocumentEvent e) {
            act();
        }
        public void insertUpdate(DocumentEvent e) {
            act();
        }
        public void act(){
            Tge1.setText(Tamb.getText());
        }
    });
}
```

```
@SuppressWarnings("unchecked")
private void initComponents() {
    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();
    jLabel6 = new javax.swing.JLabel();
    jLabel7 = new javax.swing.JLabel();
    jLabel8 = new javax.swing.JLabel();
    R = new javax.swing.JComboBox();
    combustivel = new javax.swing.JComboBox();
    quantidadeDeCaO = new javax.swing.JTextField();
    dp = new javax.swing.JTextField();
    Tamb = new javax.swing.JTextField();
    Tps1 = new javax.swing.JTextField();
    calcular = new javax.swing.JButton();
    Tge1 = new javax.swing.JFormattedTextField();
    e = new javax.swing.JFormattedTextField();
    jLabel9 = new javax.swing.JLabel();
    jLabel10 = new javax.swing.JLabel();
    jLabel11 = new javax.swing.JLabel();
    jLabel12 = new javax.swing.JLabel();
    jLabel13 = new javax.swing.JLabel();
    jLabel14 = new javax.swing.JLabel();
    jLabel15 = new javax.swing.JLabel();
    jLabel16 = new javax.swing.JLabel();
    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    jLabel1.setText("Quantidade diária de CaO a produzir");
    jLabel2.setText("Raio do Forno (interno)");
    jLabel3.setText("Diâmetro da Pedra de CaCO3");
    jLabel4.setText("Temperatura do meio ambiente");
    jLabel5.setText("Temperatura do ar à entrada do forno");
    jLabel6.setText("Temperatura da pedra de CaO à saída");
    jLabel7.setText("Combustível");
    jLabel8.setText("Excesso de ar na combustão");
    R.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "1", "1.5" }));
    combustivel.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Gas Natural" }));
}
```

```
quantidadeDeCaO.setToolTipText("10000<x<30000");
dp.setToolTipText("30<x<100");
Tamb.setToolTipText("20<x<25");
Tps1.setToolTipText("250<x<350");
calcular.setText("Calcular");
calcular.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        calcularActionPerformed(evt);
    }
});
Tge1.setFormatterFactory(new javax.swing.text.DefaultFormatterFactory(new
javax.swing.text.NumberFormatter(new java.text.DecimalFormat("#0.0"))));
Tge1.setEnabled(false);
e.setFormatterFactory(new javax.swing.text.DefaultFormatterFactory(new
javax.swing.text.NumberFormatter(new java.text.DecimalFormat("#0.00"))));
e.setToolTipText("0.2<x<0.5");
e.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        eActionPerformed(evt);
    }
});
jLabel9.setText("kg");
jLabel10.setText("m");
jLabel11.setText("mm");
jLabel12.setText("°C");
jLabel13.setText("°C");
jLabel14.setText("°C");
jLabel15.setText("%");
jLabel16.setFont(new java.awt.Font("Tahoma", 0, 8));
jLabel16.setText("(preferível usar valores médios)");
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel9)
                .addComponent(jLabel10)
                .addComponent(jLabel11)
                .addComponent(jLabel12)
                .addComponent(jLabel13)
                .addComponent(jLabel14)
                .addComponent(jLabel15)
                .addComponent(jLabel16))
            .addContainerGap(10, Short.MAX_VALUE))
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jLabel16)
            .addContainerGap(10, Short.MAX_VALUE))
);

```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jLabel2)
    .addComponent(jLabel7)
    .addComponent(jLabel1)
    .addComponent(jLabel3)
    .addComponent(jLabel4)
    .addComponent(jLabel5)
    .addComponent(jLabel6)
    .addComponent(jLabel8))
    .addGap(21, 21, 21)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
        .addComponent(Tps1, javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(R, javax.swing.GroupLayout.Alignment.LEADING, 0, 122,
Short.MAX_VALUE)
        .addComponent(quantidadeDeCaO, javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(dp, javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(Tamb, javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(combustivel, javax.swing.GroupLayout.Alignment.LEADING, 0, 1,
Short.MAX_VALUE)
        .addComponent(Tge1)
        .addComponent(e, javax.swing.GroupLayout.Alignment.LEADING))
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel9)
                .addComponent(jLabel10)
                .addComponent(jLabel11)
                .addComponent(jLabel12)
                .addComponent(jLabel13)
                .addComponent(jLabel14)))
            .addGroup(layout.createSequentialGroup()
                .addComponent(jLabel15)))
        .addGroup(layout.createSequentialGroup()
            .addComponent(jLabel16)
            .addGap(0, 0, Short.MAX_VALUE)))
    .addContainerGap())
```

```
.addGroup(layout.createSequentialGroup())
    .addGap(147, 147, 147)
    .addComponent(calcular)
    .addGap(0, 0, Short.MAX_VALUE))
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel1)
            .addComponent(quantidadeDeCaO, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jLabel9))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel2)
            .addComponent(R, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jLabel10))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel3)
            .addComponent(dp, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jLabel11))
        .addGap(1, 1, 1)
        .addComponent(jLabel16, javax.swing.GroupLayout.PREFERRED_SIZE, 10,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel4)
            .addComponent(Tamb, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jLabel12))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel5)
```

```
.addComponent(Tge1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jLabel13))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel6)
.addComponent(Tps1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jLabel14))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel7)
.addComponent(combustivel, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel8)
.addComponent(e, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jLabel15))
.addGap(18, 18, 18)
.addComponent(calcular)
.addContainerGap(18, Short.MAX_VALUE))
);
pack();
}

private void calcularActionPerformed(java.awt.event.ActionEvent evt) {
    if(validaDadosUtilizador()){
        setDadosUtilizador();
        Resultados.resultados = new StringBuilder();
        Equations.equation1();
        Fase2.Equations.equation2();
        Fase3.Equations.equation3();
        Fase4.Equations.equation4();
        Fase5.Equations.equation5();
        if(Fase6.Equations.equation6() == true){
            if(Fase7.Equations.equation7() == true){
                Fase8.Equations.equation8();
            }
        }
    }
}
```



```
Fase9.Equations.equation9();
Fase91.Equations.equation91();
chamarResultados();
}
else{
    JOptionPane.showMessageDialog(null, "Erro de cálculo: Condição convectiva não
cumprida (placa vertical).");
}
}
else{
    JOptionPane.showMessageDialog(null, "Erro de cálculo: Altura do forno insuficiente");
}
}
else{
    JOptionPane.showMessageDialog(null, this.error);
}
}
private void eActionPerformed(java.awt.event.ActionEvent evt) {
}
private boolean validaDadosUtilizador(){
    return validaInputUtilizador() && validaIntervalos();
}
private boolean validaInputUtilizador(){
    String quantidadeDeCaO_aux = quantidadeDeCaO.getText();
    String R_aux = R.getSelectedItemAt().toString();
    String dp_aux = dp.getText();
    dp_aux = dp_aux.replaceAll(",", ".");
    String Tamb_aux = Tamb.getText();
    String Tge1_aux = Tge1.getText();
    String Tps1_aux = Tps1.getText();
    String e_aux = e.getText();
    e_aux = e_aux.replaceAll(",", ".");
    //String combustivel_aux = combustivel.getSelectedItemAt().toString();
    try{
        this.quantidadeDeCaO_auxInt = Integer.parseInt(quantidadeDeCaO_aux);
        this.quantidadeDeCaCO3_auxDouble = this.quantidadeDeCaO_auxInt * (0.1/0.056);
    } catch (NumberFormatException e){this.error = "Caracter inválido em: Quantidade De CaO";
return false;}
```

```
this.R_auxDouble = Double.parseDouble(R_aux);
try{
    this.dp_auxDouble = Double.parseDouble(dp_aux);
} catch(NumberFormatException e){this.error = "Caracter inválido em: Diâmetro da pedra de
CaO3"; return false;}
try{
    this.Tamb_auxInt = Integer.parseInt(Tamb_aux);
} catch(NumberFormatException e){this.error = "Caracter inválido em: Temperatura do meio
ambiente"; return false;}
try{
    this.Tge1_auxInt = Integer.parseInt(Tge1_aux);
} catch(NumberFormatException e){this.error = "Caracter inválido em: Temperatura do ar à
entrada"; return false;}
try{
    this.Tps1_auxInt = Integer.parseInt(Tps1_aux);
} catch(NumberFormatException e){this.error = "Caracter inválido em: Temperatura da pedra à
saída"; return false;}
try{
    this.e_auxDouble = Double.parseDouble(e_aux);
} catch(NumberFormatException e){this.error = "Caracter inválido em: Excesso de ar"; return
false;}
this.error = "";
return true;
}
private boolean validaIntervalos(){
    if(quantidadeDeCaO_auxInt < 10000 || quantidadeDeCaO_auxInt > 30000){
        this.error = "Valor inválido (" + quantidadeDeCaO_auxInt + ") em: Quantidade diária de CaO a
produzir [10000;30000]";
        return false;
    }
    else if(dp_auxDouble < 30.0 || dp_auxDouble > 100.0){
        this.error = "Valor inválido (" + dp_auxDouble + ") em: Diâmetro da pedra de CaCO3 [30;100]";
        return false;
    }
    else if(Tamb_auxInt < 20 || Tamb_auxInt > 25){
        this.error = "Valor inválido (" + Tamb_auxInt + ") em: Temperatura do meio ambiente [20;25]";
        return false;
    }
}
```

```
else if(Tps1_auxInt < 250 || Tps1_auxInt > 350){
    this.error = "Valor inválido (" + Tps1_auxInt + ") em: Temperatura da pedra à saída [250;350]";
    return false;
}
else if(e_auxDouble < 0.2 || e_auxDouble > 0.5){
    this.error = "Valor inválido (" + e_auxDouble + ") em: Excesso de ar [0.2;0.5]";
    return false;
}
else{
    this.error = "";
    return true;
}
}

private void setDadosUtilizador(){
    Variables.quantidadeDeCaO = quantidadeDeCaO_auxInt;
    Variables.quantidadeDeCaCO3 = quantidadeDeCaCO3_auxDouble;
    Variables.dp = dp_auxDouble;
    Variables.R = R_auxDouble;
    Fase3.Variables.Tge1 = Tge1_auxInt;
    Fase3.Variables.Tps1 = Tps1_auxInt;
    Fase4.Variables.e = e_auxDouble;
    Fase7.Variables.Tamb = Tamb_auxInt;
    Fase8.Variables.alpha = 0.25;
}

public void chamarResultados(){
    new Resultados().setVisible(true);
}

public static void main(String args[]) {
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    }
    catch (ClassNotFoundException ex) {
```

```
java.util.logging.Logger.getLogger(Principal.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (InstantiationException ex) {
java.util.logging.Logger.getLogger(Principal.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (IllegalAccessException ex) {
java.util.logging.Logger.getLogger(Principal.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
java.util.logging.Logger.getLogger(Principal.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    }
}
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new Principal().setVisible(true);
    }
});
}
// Variables declaration - do not modify
private javax.swing.JComboBox R;
private javax.swing.JTextField Tamb;
private javax.swing.JFormattedTextField Tge1;
private javax.swing.JTextField Tps1;
private javax.swing.JButton calcular;
private javax.swing.JComboBox combustivel;
private javax.swing.JTextField dp;
private javax.swing.JFormattedTextField e;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel14;
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel16;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
```

```
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JTextField quantidadeDeCaO;
}

package Viewer;
import java.awt.Dimension;
import java.awt.Toolkit;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
public class Resultados extends javax.swing.JFrame {
    public static StringBuilder resultados = new StringBuilder();
    private final JFileChooser fc;
    public Resultados() {
        this.setTitle("FornoCal - Resultados");
        initComponents();
        setResultados();
        this.fc = new JFileChooser();
        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        this.setResizable(false);
        Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
        Dimension frameSize = this.getSize();
        if (frameSize.height > screenSize.height) {
            frameSize.height = screenSize.height;
        }
        if (frameSize.width > screenSize.width) {
            frameSize.width = screenSize.width;
        }
        this.setLocation( (screenSize.width - frameSize.width)/2, (screenSize.height -
frameSize.height)/2);
    }
    public void setResultados(){
```

```
this.H.setText(Double.toString(round(Fase1.Variables.H, 3)));
this.H1.setText(Double.toString(round(Fase3.Variables.H1, 3)));
this.H2.setText(Double.toString(round(Fase2.Variables.H2, 3)));
this.H3.setText(Double.toString(round(Fase5.Variables.H3, 3)));
this.D.setText(Double.toString(round(Fase7.Variables.D, 3)));
this.esp.setText(Double.toString(round(Fase7.Variables.esp, 3)));
this.Tpex.setText(Double.toString(round(Fase7.Variables.Tpex, 3)));
this.Tpex3p.setText(Double.toString(round(Fase91.Variables.Tpex3p, 3)));
this.mComb.setText(Double.toString(round(Fase4.Variables.mComb, 3)));
this.DeltaP.setText(Double.toString(round(Fase8.Variables.DeltaP, 3)));
}

public static double round(double value, int places) {
    if (places < 0)
        throw new IllegalArgumentException();
    long factor = (long) Math.pow(10, places);
    value = value * factor;
    long tmp = Math.round(value);
    return (double) tmp / factor;
}

@SuppressWarnings("unchecked")
private void initComponents() {
    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();
    jLabel6 = new javax.swing.JLabel();
    jLabel7 = new javax.swing.JLabel();
    jLabel8 = new javax.swing.JLabel();
    jLabel9 = new javax.swing.JLabel();
    jLabel10 = new javax.swing.JLabel();
    H = new javax.swing.JTextField();
    H1 = new javax.swing.JTextField();
    H2 = new javax.swing.JTextField();
    D = new javax.swing.JTextField();
    esp = new javax.swing.JTextField();
    Tpex = new javax.swing.JTextField();
    Tpex3p = new javax.swing.JTextField();
}
```

```
mComb = new javax.swing.JTextField();
DeltaP = new javax.swing.JTextField();
Imprimir = new javax.swing.JButton();
H3 = new javax.swing.JTextField();
jLabel11 = new javax.swing.JLabel();
jLabel12 = new javax.swing.JLabel();
jLabel13 = new javax.swing.JLabel();
jLabel14 = new javax.swing.JLabel();
jLabel15 = new javax.swing.JLabel();
jLabel16 = new javax.swing.JLabel();
jLabel17 = new javax.swing.JLabel();
jLabel18 = new javax.swing.JLabel();
jLabel19 = new javax.swing.JLabel();
jLabel20 = new javax.swing.JLabel();
jLabel21 = new javax.swing.JLabel();
jLabel22 = new javax.swing.JLabel();
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
jLabel1.setText("Altura do Forno");
jLabel2.setText("Zona de Arrefecimento");
jLabel3.setText("Zona de Calcinação");
jLabel4.setText("Zona de pré-aquecimento");
jLabel5.setText("Diâmetro do Forno (externo)");
jLabel6.setText("Espessura da parede");
jLabel7.setText("Temperatura da parede externa");
jLabel8.setText("Temperatura da parede externa");
jLabel9.setText("Caudal de Combustível");
jLabel10.setText("Perda de carga");
H.setEnabled(false);
H.setMaximumSize(new java.awt.Dimension(6, 20));
H1.setEnabled(false);
H2.setEnabled(false);
D.setEnabled(false);
D.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        DActionPerformed(evt);
    }
});
esp.setEnabled(false);
```

```
    Tpex.setEnabled(false);
    Tpex3p.setEnabled(false);
    mComb.setEnabled(false);
    DeltaP.setEnabled(false);
    Imprimir.setText("Imprimir");
    Imprimir.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            ImprimirActionPerformed(evt);
        }
    });
    H3.setEnabled(false);
    jLabel11.setFont(new java.awt.Font("Tahoma", 0, 8));
    jLabel11.setText("(Zona de Calcinação)");
    jLabel12.setFont(new java.awt.Font("Tahoma", 0, 8));
    jLabel12.setText("(Zona de pré-Aquecimento)");
    jLabel13.setText("m");
    jLabel14.setText("m");
    jLabel15.setText("m");
    jLabel16.setText("m");
    jLabel17.setText("m");
    jLabel18.setText("°C");
    jLabel19.setText("°C");
    jLabel20.setText("Pa");
    jLabel21.setText("kg/s");
    jLabel22.setText("m");
    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .add(layout.createSequentialGroup()
                        .add(layout.createSequentialGroup()
                            .addContainerGap()
                            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                .add(layout.createSequentialGroup()
                                    .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                        .add(layout.createSequentialGroup()
                                            .addComponent(jLabel1)
                                            .addComponent(jLabel5)
                                            .addComponent(jLabel6)
                                            .addComponent(jLabel7)
```



```

        .addComponent(jLabel8)
        .addComponent(jLabel9)
        .addComponent(jLabel10)
        .addGroup(layout.createSequentialGroup()
            .addGap(32, 32, 32)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLabel3)
            .addComponent(jLabel2)
            .addComponent(jLabel4))))
        .addGap(29, 29, 29)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
        .addComponent(H1)
        .addComponent(H2)
        .addComponent(D)
        .addComponent(esp)
        .addComponent(Tpex)
        .addComponent(Tpex3p)
        .addComponent(mComb)
        .addComponent(DeltaP, javax.swing.GroupLayout.DEFAULT_SIZE, 72,
Short.MAX_VALUE)
        .addComponent(H3)
        .addComponent(H, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jLabel13)
        .addComponent(jLabel14)
        .addComponent(jLabel15)
        .addComponent(jLabel16)
        .addComponent(jLabel17)
        .addComponent(jLabel18)
        .addComponent(jLabel19)
        .addComponent(jLabel20)
        .addComponent(jLabel21)
        .addComponent(jLabel22)))
        .addComponent(jLabel11)
        .addComponent(jLabel12))
    .addContainerGap(24, Short.MAX_VALUE))

```

```
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup())
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(Imprimir)
    .addGap(123, 123, 123))
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel1)
            .addComponent(H, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jLabel13))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel2)
            .addComponent(H1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jLabel14))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel3)
            .addComponent(H2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jLabel15))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel4)
            .addComponent(H3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jLabel16))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel5)
            .addComponent(D, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jLabel17))
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel6)
    .addComponent(esp, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel22))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel7)
    .addComponent(Tpex, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel18))
.addGap(1, 1, 1)
.addComponent(jLabel11)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel8)
    .addComponent(Tpex3p, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel19))
.addGap(1, 1, 1)
.addComponent(jLabel12)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jLabel9)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(mComb, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel21)))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jLabel10)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(DeltaP, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel20)))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 16, Short.MAX_VALUE)
.addComponent(Imprimir)
```

```
.addContainerGap()
);
pack();
}
private void ImprimirActionPerformed(java.awt.event.ActionEvent evt) {
    guardar_ficheiroActionPerformed(evt);
}
private void guardar_ficheiroActionPerformed(java.awt.event.ActionEvent evt) {
    //this.erro_nao_existe.setVisible(false);
    int returnVal = this.fc.showSaveDialog(this);
    if (returnVal == JFileChooser.APPROVE_OPTION) {
        File file = this.fc.getSelectedFile();
        try {
            if(file.exists() == false){
                if(guardarFicheiro(file.getPath()) == 0){
                    JOptionPane.showMessageDialog(null, "Erro na escrita do ficheiro de resultados!
Tente novamente.");
                }
            }
            else {
                JOptionPane.showMessageDialog(null, "Já existe um ficheiro com esse nome! Tente
novamente.");
            }
        } catch (IOException ex) {
            JOptionPane.showMessageDialog(null, "Erro na escrita do ficheiro de resultados! Tente
novamente.");
        }
    } else {
        JOptionPane.showMessageDialog(null, "Nada foi escrito! Tente novamente.");
    }
}
private int guardarFicheiro(String caminho_ficheiro_guardar) throws IOException {
    if(caminho_ficheiro_guardar != null){
        BufferedWriter out = new BufferedWriter(new FileWriter(caminho_ficheiro_guardar));
        out.write(this.resultados.toString());
        out.close();
        return 1;
    }
}
```

```
    else{
        return 0;
    }
}

private void DActionPerformed(java.awt.event.ActionEvent evt) {
}

private javax.swing.JTextField D;
private javax.swing.JTextField DeltaP;
private javax.swing.JTextField H;
private javax.swing.JTextField H1;
private javax.swing.JTextField H2;
private javax.swing.JTextField H3;
private javax.swing.JButton Imprimir;
private javax.swing.JTextField Tpex;
private javax.swing.JTextField Tpex3p;
private javax.swing.JTextField esp;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel14;
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel16;
private javax.swing.JLabel jLabel17;
private javax.swing.JLabel jLabel18;
private javax.swing.JLabel jLabel19;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel20;
private javax.swing.JLabel jLabel21;
private javax.swing.JLabel jLabel22;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
```

```
private javax.swing.JTextField mComb;  
}
```

Anexo B

Propriedades físicas

Polinómios desenvolvidos para o programa das propriedades físicas do ar à pressão atmosférica.

- Condutividade térmica:

$$k(x) = -1,1198897E-17x^6 + 8,7864215E-15x^5 - 4,4374479E-12x^4 + 9,6559189E-10x^3 - 6,7839993E-08x^2 + 7,1477994E-05x + 2,4295037E-02$$

- Viscosidade cinemática:

$$\nu(x) = 4,365140866895E-20x^6 - 4,039290616056E-17x^5 + 1,088015085567E-14x^4 - 1,141636784366E-12x^3 + 1,444436360192E-10x^2 + 8,699748434118E-08x + 1,331422047548E-05$$

- Coeficiente de expansão volumétrica:

$$\beta(x) = 5,8070940E-17x^6 - 2,9994199E-14x^5 + 5,1941928E-12x^4 - 3,7374860E-10x^3 + 3,9708796E-08x^2 - 1,2953005E-05x + 3,6734906E-03$$

- Número de Prandtl:

$$Pr(x) = -8,9663606E-15x^6 + 4,2151568E-12x^5 - 5,8250457E-10x^4 + 8,6537272E-09x^3 + 2,4053808E-06x^2 - 1,9350745E-04x + 7,1547018E-01$$

Anexo C

Tabelas das propriedades físicas do tijolo refratário e isolante.